SigC5561 Reference Guide

Copyright © 2006-2007 Signalogic, Inc.
Revision A6
Jul 2007

Signalogic, Inc.
9617 Wendell @ Skillman
Dallas, TX 75243
Tel: 214-349-5551
Fax: 214-343-0163

http://www.signalogic.com
tech_support@signalogic.com

TABLE OF CONTENTS

1 System Overview 7
1.1 Documentation Notes and Conventions 10
1.2 Additional Documentation Sources 10
1.3 Contacting Signalogic 11
1.3.1 Technical Support Procedures 11
1.3.2 Frequently Asked Questions 11
1.3.3 Returning Merchandise for Service 12
1.4 Hardware Overview 13
1.5 Software Overview 14
2 Hardware Installation 16
2.1 Hardware Checklist 17
2.1.1 Functional Test Hardware Checklist 17
2.2 Installation Procedure 18
2.2.1 System Preparation 18
2.2.2 Anti-Static Precautions and Handling 18
2.2.3 Unpacking 20
2.2.4 Visual Inspection 20
2.2.5 Installing the SigC5561 PCI Card 21
2.2.6 Installing the SigC5561-PTMC Module 21
2.2.7 GbE and 10/100 Ports 21
2.2.8 Connecting Cables 21
2.2.9 FTP Server 21
2.3 Power-On 21
2.3.1 Panel LED Indicators at Power-On 21
2.3.2 Status / Error Code Indicators at Power-On 21
2.3.3 Verifying PCI Status 21
3 Software Installation 21
3.1 Software Checklist 21
3.1.1 Signalogic Software Checklist 21
3.1.2 Functional Test Software Checklist 21
3.1.3 Directory Structure 21
3.1.4 Electronic Format Software Upgrade 21
3.1.5 Supported Linux OS Installation 21
3.2 Software Setup and Configuration 21
3.2.1 Board Configuration File 21

3.2.2 FTP Server Setup 21
3.2.3 SigC5561 Linux Driver 21
3.2.4 Ethereal Installation 21
4 Test and Checkout 21
4.1 API Functional Test 21
4.2 PCI Terminal Program 21
4.3 Ethereal Packet Verification 21
4.4 Troubleshooting 21
5 Platform Support and Configuration 21
5.1 Boot Configuration 21
5.1.1 FTP Download 21
5.1.2 Flash Load 21
5.2 Platform Support 21
5.2.1 PCI Platform Support 21
5.2.2 cPCI Platform Support 21
5.2.3 ATCA Platform Support 21
5.2.4 Debug Tab 21
5.2.5 Stand-Alone Configuration 21
6 Building Host Application Software 21
6.1 Rebuilding SigC5561 LKM Driver 21
6.2 Rebuilding Test Programs 21
6.2.1 Rebuilding the SigC5561 Terminal Emulation Program 21
6.3 TDM-to-IP Testing 21
6.4 IP-to-IP Testing 21
6.5 Conferencing Example 21
6.6 Asterisk SIP Proxy Example 21
7 Card Resources and Peripherals 21
7.1 Host Interface 21
7.1.1 PCI Host Interface 21
7.1.1.1 PCI Configuration Space Values 21
7.1.1.2 BAR0 Description 21
7.1.1.3 BAR1 Description 21
7.1.1.4 BAR2 Description 21
7.2 DSP HPI Register Access 21

7.3 Logic Registers 21
7.4 TDM Registers 21
7.5 Flash 21
7.5.1 Boot Flash Configuration 21
7.5.2 Network Processor Code Flash Configuration 21
7.6 LEDs 21
7.6.1 Panel LED Indicators 21
7.6.2 Status / Error Code LED Indicators 21
7.7 DMA Controller 21
7.7.1 DMA Controller Testing 21
7.7.2 Initial DMA Configuration 21
7.7.3 DMA Controller Performance 21

 Copyright © Signalogic 2006-2007

Revision History

Created	Rev A	Mar 2007 Consolidated from separate documents
Revised	Rev A1	May 2007
Revised	Rev A2	May 2007, corrections to API functional test section
Revised	Rev A3	May 2007, additions to test and checkout procedures
Revised	Rev A4	May 2007, test and checkout additions, cont.
Revised	Rev A5	Jun 2007, Chapter 6 additions
Revised	Rev A6	Jul 2007, Asterisk and SIP proxy example

 Copyright © Signalogic 2006-2007

This page intentionally left blank.

1 System Overview

This document assumes you are using an x86-compatible system, running Linux, and you will install and operate the SigC5561 PCI card or SigC5561-PTMC module. Currently tested configurations include:

OS	Kernel
Red Hat	2.4.18, 2.6.9
CentOS	2.4.18, 2.6.9

For the SigC5561 PCI card, in order to run the software and programs described in this guide, a standard desktop or server should be available as the host, or "target" system. Some examples are shown in Figures 1-1 through 1-3, below.

Figure 1-1 *Desktop Configuration*

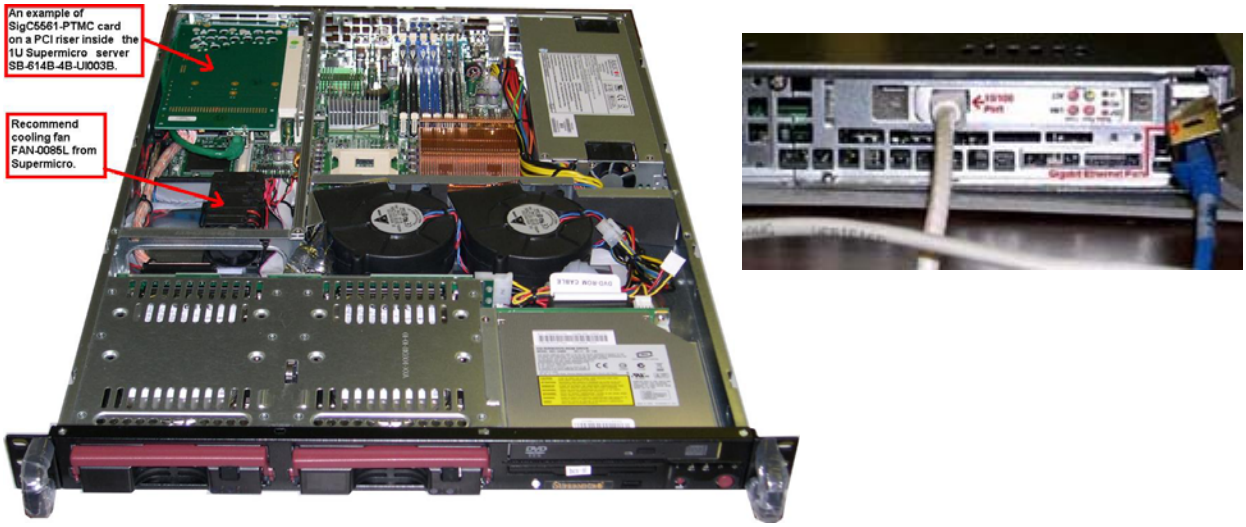
An example of SigC5561-PTMC card on a PCI riser inside the 1U Supermicro server SB-614B-4B-UI003B.
Recommend cooling fan FAN-0085L from Supermicro.

Figure 1-2 *1U Server Configuration*

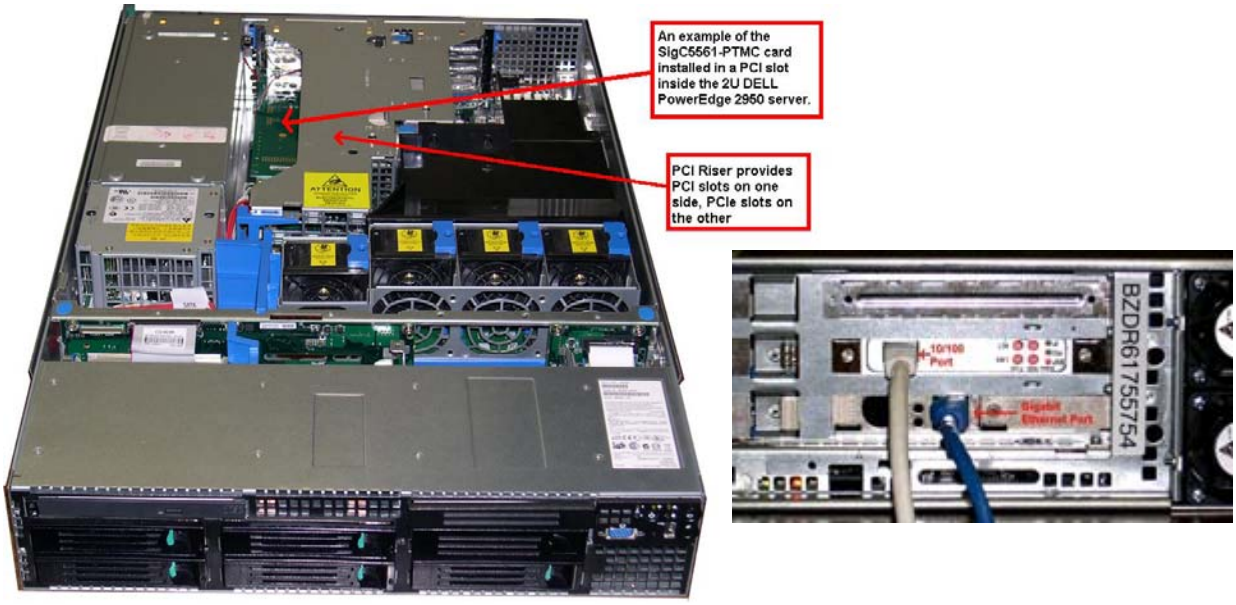
An example of the SigC5561-PTMC card installed in a PCI slot inside the 2U DELL PowerEdge 2950 server.
PCI Riser provides PCI slots on one side, PCIe slots on the other
BZDR61755754

Figure 1-3 *2U Server Configuration*

For the SigC5561-PTMC module, in order to run the software and programs described in this guide, a cPCI or ATCA chassis should be available as the host, or "target" system. An example is shown in the Figure 1-4 below.

Figure 1-4 *cPCI Chassis Configuration*

1.1 Documentation Notes and Conventions

Below are some common conventions used in this documentation.

1) Command line examples in the following sections beginning with

```
shell/>
```

indicate a command that needs to be typed in at the shell command prompt.

2) “WinXP” refers also to Win2K systems.

3) The words “board” and “card” are used interchangeably to refer to DSP based electronic circuit boards. Generally, the word card is used when there is a possibility of confusing the DSP card with a motherboard or carrier board. In that case, the word card is used in the sense of “daughtercard”.

4) GbE is used to refer to Gigabit Ethernet.

5) The nomenclature

```
..\
```

is used to refer to software installation subfolders in some cases. For example, to refer to .obj files in shorthand and include both C5xxx and C6xxx hardware, the documentation might say “.obj files are located on ..\SCI\cStandard subfolders” which means .obj files can be found on

```
\dspcode\c5xxx\SCI\cStandard
```

subfolder for C5xxx hardware, and on

```
\dspcode\c6xxx\SCI\cStandard
```

subfolder for C6xxx hardware.

6) Linux machine screen captures shown throughout this documentation were obtained from a CentOS system, using GUI terminal windows. GUI terminals are used in most examples in this document in order to provide scroll bars and white background with black text suitable for easy-to-read screen captures.

1.2 Additional Documentation Sources

The C5x/C6x Reference Guide is a more general document covering various types of Signalogic hardware, and is located online at:

ftp://ftp.signalogic.com/documentation/Boards/SigC55xx

and at

ftp://ftp.signalogic.com/documentation/Boards/SigC64xx

A server / system level guide for using the SigC5561 card, "System Recommendations for SigC5561 Card Usage", can be found at:

ftp://ftp.signalogic.com/documentation/Boards/SigC55xx

1.3 Contacting Signalogic

Address: Signalogic, Inc.
9617 Wendell @ Skillman
Dallas, TX 75243
USA

Tel: +1 (214) 349-5551
Fax: +1 (214) 343-0163

Email: tech_support@signalogic.com
dspinfo@signalogic.com

Web Site: www.signalogic.com

1.3.1 Technical Support Procedures

Before contacting Signalogic technical support, please take the following steps:

- Check section 4.4, Troubleshooting
- Please go through the "Frequently Asked Question" (FAQ) sections in this guide or see the FAQs on our web site:

  http://www.signalogic.com/index.pl?page=voip_faq

If the problem is not resolved, then send e-mail to tech_support@signalogic.com that includes a detailed problem description and a subject line with the following format:

```
[SigC5561] customer name, problem summary
```

1.3.2 Frequently Asked Questions

Frequently Asked Questions can be found at:

http://www.signalogic.com/index.pl?page=voip_faq

 Copyright © Signalogic 2006-2007

1.3.3 Returning Merchandise for Service

A receipt or copy of your invoice marked with the date of purchase is required before any warranty service will be rendered. You can obtain service by calling Signalogic for a Returned Merchandise Authorization (RMA) number. When returning to the manufacture, the RMA number should be prominently displayed on the outside of the shipping box, and mailed prepaid or hand-carried.

This warranty only covers normal consumer use and does not cover damages incurred in shipping or from failure due to the alternation, misuse, abuse or improper maintenance of products.

During the warranty period, contact Signalogic for any product problems.

 Copyright © Signalogic 2006-2007

1.4 Hardware Overview

This document expects that you have one of the hardware products listed below:

SigC5561 PCI Card

- PCI 2.2 compliant, half-length, full-height PCI card
- 3.3V PCI slot only
- Intended for PC / server use
- Compatible with telephony PCI cards (FXS/FXO, quad T1/E1, etc.)

SigC5561-PTMC Module

- PMC form-factor module supporting PTMC electrical configurations 2, 3, and 5
- 3.3V PMC site only
- Intended for cPCI or ATCA carrier (blade) use
- Compatible with T3 and OC-3 PTMC modules

 Copyright © Signalogic 2006-2007

1.5 Software Overview

Below is a summary of software tools referred to in this documentation.

Software	Provider	Layer	Overview	Comments
DirectDSP software Linux version [1], including • Hardware Library shared object module, v1.2 • Engine Manager shared object module, v1.2 • Hardware Manager shared object module, v1.2	Signalogic	Library / API	DSP Hardware Interface	Linux only
DirectDSP software WinXP version, including • Hardware Library DLL, v3.1 • Engine Manager DLL, v3.1 • Hardware Manager DLL, v3.1	Signalogic	Library / API	DSP Hardware Interface	Win9x and WinXP
DSPower-MGTS	Signalogic	Library / API	VoP extension to DirectDSP software	Provides VoIP API, provides TDM interface API
Kernel mode drivers	Signalogic	Driver	Low-level hardware interface	.sys and .inf files for Win2k and WinXP systems Loadable kernel module (LKM) driver, v1.2 for Linux [1]
GCC tools		Application	Host Application Development	Typically used to build GUI and host-based applications that will make DirectDSP API calls to interface to DSP hardware. Optional, depends on application
VxWorks	Wind River	Board-level real-time operating system	RTOS that runs on WinPath network processor	MIPS core version. Not user programmable unless by special arrangement with Signalogic

Utilities	Signalogic	Application	Hardware support, test, and diagnostic	Examples include Flash utility, terminal emulation utility, test programs, VoIP demo programs

Notes

[1] Assumes Red Hat Linux v7.3 or higher, Kernel v2.4.18-3 or higher.

 Copyright © Signalogic 2006-2007

2 Hardware Installation

Please follow the instructions in sections 2.1 and 2.2 below, depending on your specific type of hardware.

Also, the following links may be helpful depending on specific type of software or hardware that you have.

http://www.signalogic.com/index.pl?page=tech_support#Getting_Started

ftp://ftp.signalogic.com/documentation/Boards/SigC67xx/SigC67xxSetup.PDF

ftp://ftp.signalogic.com/documentation/Boards/SigC67xx/VDS_Users_Guide.pdf

ftp://ftp.signalogic.com/documentation/DirectDSP/Users 20Guide/

ftp://ftp.signalogic.com/documentation/DirectDSP/Reference 20Guide/

ftp://ftp.signalogic.com/documentation/SCI/C5xC6xCCS20JTAG.PDF

2.1 Hardware Checklist

Note: The SigC5561 product may have been tested in a customer-provided system or a turn-key system may have been ordered from Signalogic. In these cases, you may be testing a pre-installed system and can skip to section 4.0 below, Test and Checkout. If the SigC5561 PCI card or SigC5561-PTMC module has been sent separately, please continue with the delivery inspection steps listed below.

If the product is a SigC5561 PCI card, please check that the following items are included with the shipment.

1) SigC5561 PCI card configured with 2-DSP, 7-DSP, or 12-DSP option.
2) Optional 1' (12", or 30.5 cm) CAT6 Ethernet extension cable (male-to-male). Must be specified at time of order.
3) Optional software CD. Must be specified at time of order.

If the product is a SigC5561-PTMC module, then please verify the following items are included in the shipment:

1) SigC5561-PTMC module configured with 2-DSP, 7-DSP, or 12-DSP option.
2) Optional software CD. Must be specified at time of order.

If anything listed here is damaged or missing, please contact Signalogic immediately.

In all cases, the SigC5561 card or module should be enclosed in a sealed anti-static bag. If this is not the case, or you are for any reason worried that customary anti-static handling procedures have not been followed, please let Signalogic know.

2.1.1 Functional Test Hardware Checklist

Signalogic recommends the following hardware to be used during the SigC5561 card functional test and checkout procedure:

1. 10/100 Ethernet Switch
2. 10/1000 T-Base Gigabit Switch with Port Mirroring capability. There should be a minimum of four (4) GbE (10/1000) ports on the switch. The Netgear model # FS728TS is one example of such a switch
3. CAT 6 network cables

In addition it's expected that common add-in card mounting tools and hardware will be on-hand, such as screwdrivers, standard PC backplate mounting screws, etc.

2.2 Installation Procedure

The following sections cover installation and preparation of the SigC5561 PCI card or SigC5561-PTMC module.

2.2.1 System Preparation

If the host machine supports PCI bus speeds faster than 33 MHz, then make sure the BIOS setting is either 'Auto' or fixed at 33 MHz.

The SigC5561 PCI card and PTMC module products are both factory-configured via a pull-down resistor to operate at 33 MHz PCI bus rate. Some tests at Signalogic indicate the SigC5561 may work at 66 MHz rate, but this is not yet officially supported.

2.2.2 Anti-Static Precautions and Handling

ElectroStatic Discharge (ESD) can damage sensitive electronic components. To prevent damage to the SigC5561 card or SigC5561-PTMC module, it's important to handle these products very carefully. The following measures are generally sufficient to protect the SigC5561 hardware products from ESD:

1. Use a grounded wrist strap designed to prevent static discharge.
2. Touch a grounded metal object, such as the metal case or chassis of a PC, before removing the card or module from the anti-static bag. After touching the grounded metal, *do not move your feet*. Hold your feet still until the card is either installed in the computer system, or replaced within the anti-static bag.
3. Handle the PCI card or PTMC module by its edges only. Do not touch its components or gold contacts. Proper grip when handling the card or module is shown in Figure 2-1 and Figure 2-2 below.
4. Make sure your computer system chassis provides conductivity between the power supply internal case, the computer external case, mounting fasteners, and the motherboard. A typical way to verify this is to use a multimeter to measure continuity between various points in your computer system. *Always do this with the computer system turned off and all AC power plugs completely disconnected from the system.*

 Copyright © Signalogic 2006-2007

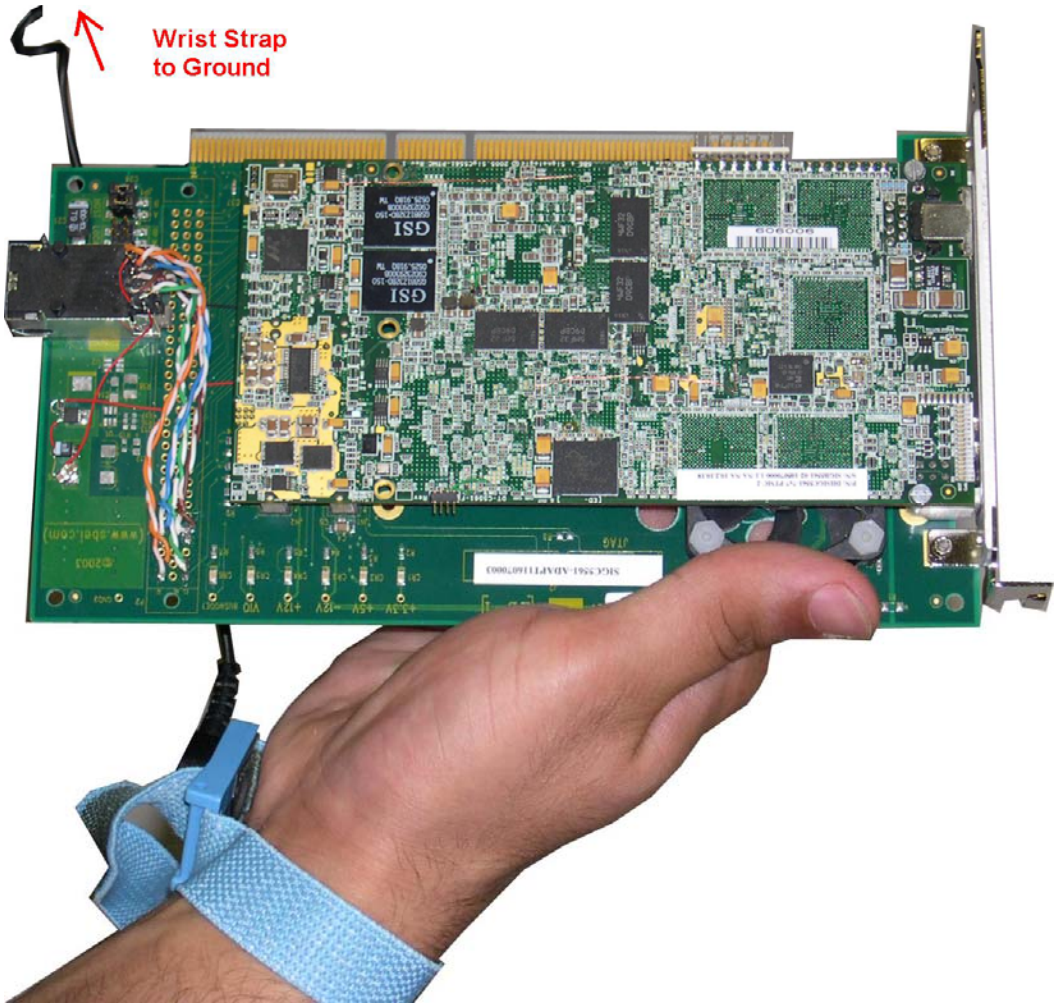

Figure 2-1 *Proper Grip for Handling PCI Card*

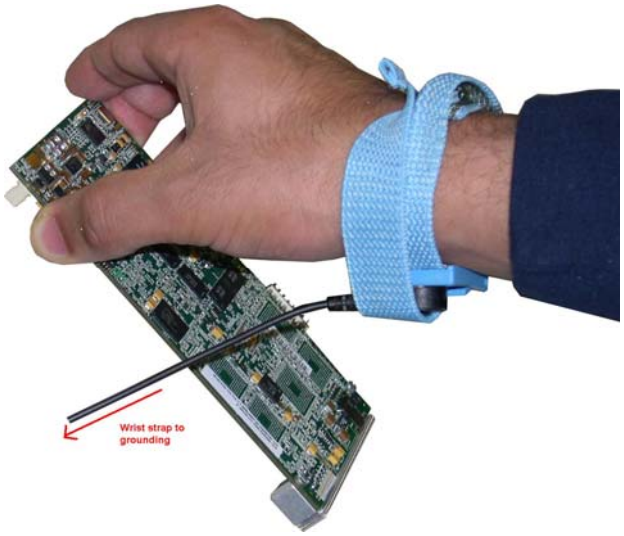

Figure 2-2 *Proper Grip for Handling PTMC Module*

In addition to ESD concerns, the SigC5561 products contain many ultra-small components, including some known as “0201”, a term which describes a component package size commonly found in cell phones, PDAs, media players, and other extremely dense electronic products. It’s imperative that you *avoid touching any part of the surface* of the SigC5561 card or module in order to prevent cracking or otherwise damaging these small components.

2.2.3 Unpacking

SigC5561 PCI cards and SigC5561-PTMC modules are shipped in anti-static packaging to avoid ESD. When unpacking a card or module, please proceed slowly and deliberately. Carefully pull off the anti-static sticker that seals the packaging. If the card or module seems to be sticking to its packaging, take your time and gently unstick it. Above all, be sure the person handling it is static-safe -- see section 2.2.1 above, Anti-Static Precautions and Handling.

2.2.4 Visual Inspection

Upon unpacking the SigC5561 PCI card, or SigC5561-PTMC module products, they should appear as shown in Figure 2-3 and Figure 2-4 below, respectively. To inspect the product for shipping and handling damage, go through the following checklist:

1. Check around connectors looking for bending or dents
2. Look for “can-style” capacitors that may be bent or twisted
3. Look for mounting screws that may have come loose
4. Look for scratches or scrape markings on the PCB surface
5. Look for any loose pieces of metal or other debris which may have somehow entered the packaging and anti-static bag

At no time should you touch or poke at components or jumper wires on the card. ***Visually inspect only !!***

If you find any of the above anomalies, please report your findings to Signalogic tech support immediately. You may be asked to provide a digital picture of what you found, to assist tech support personnel in determining whether it’s a significant issue and corrective action is warranted.

 Copyright © Signalogic 2006-2007

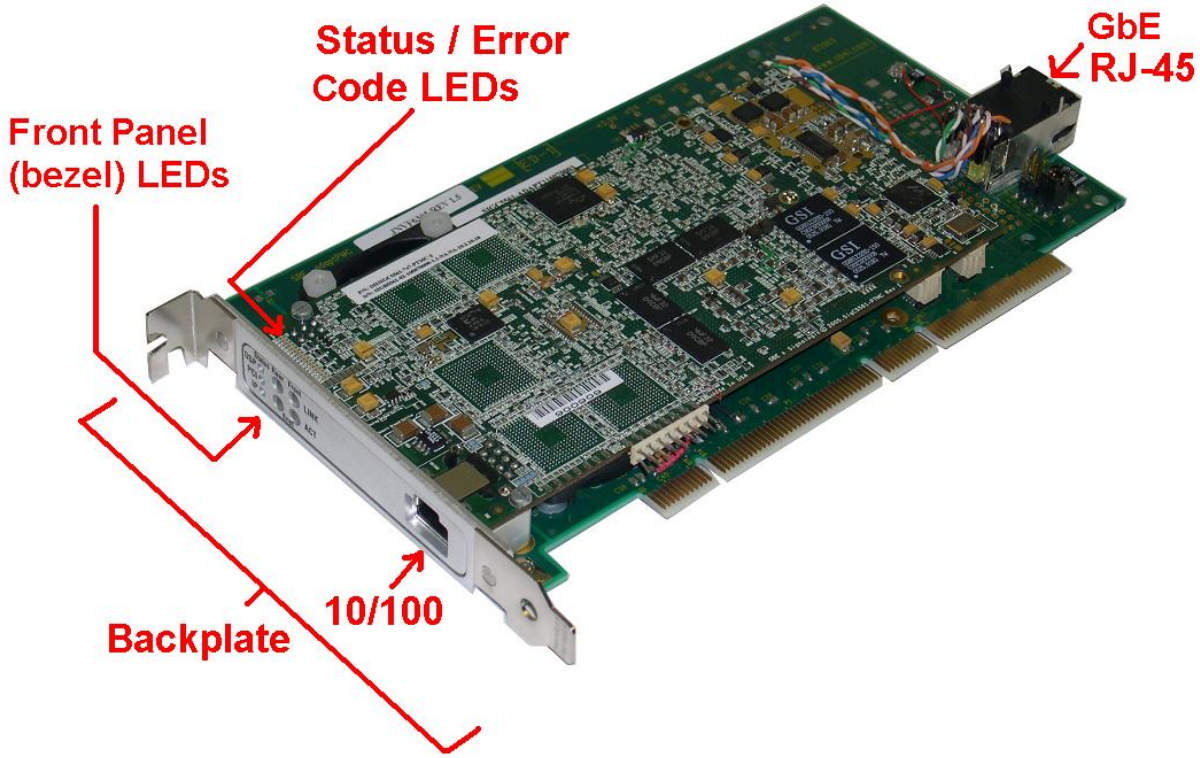

Figure 2-3 *SigC5561 PCI Card*

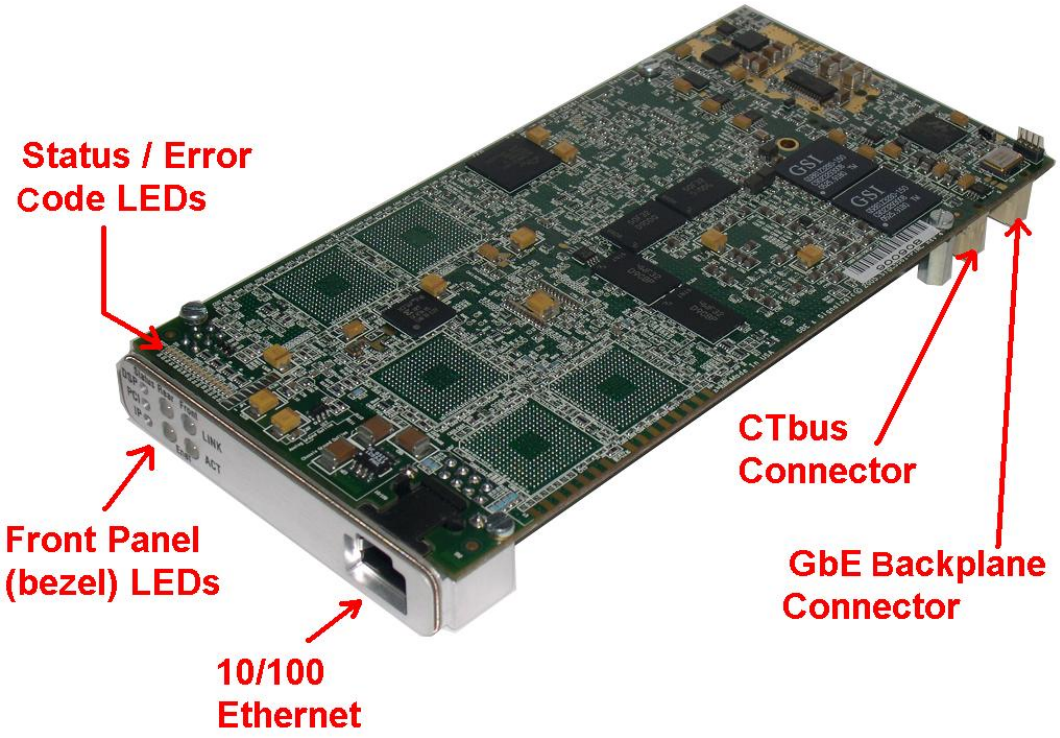

Figure 2-4 *SigC5561-PTMC Module*

2.2.5 Installing the SigC5561 PCI Card

CAUTION Exercise extreme care when installing or removing the SigC5561 PCI card to and from computer systems to prevent any possible damage. Please handle the card gently and avoid contact with any fluids or organic material.

Note: First make sure power is completely off to the PC or server. There should be no LED indicators of any type glowing on the motherboard. It's advisable to disconnect all AC power plugs from the PC or server. If you plug the SigC5561 PCI card into a PC with power still enabled, you will damage the card and void its warranty.

Note: Only install the SigC5561 card into a 3.3 V PCI slot !! If you plug the SigC5561 card into a 5V PCI slot, you will damage the card and its warranty will become void!

To install the SigC5561 card, follow the steps below:

1) Insert the SigC5561 card into a 3.3V PCI slot on the motherboard of the PC or server. An example of a 3.3V PCI slot is shown in Figure 2-5 below.

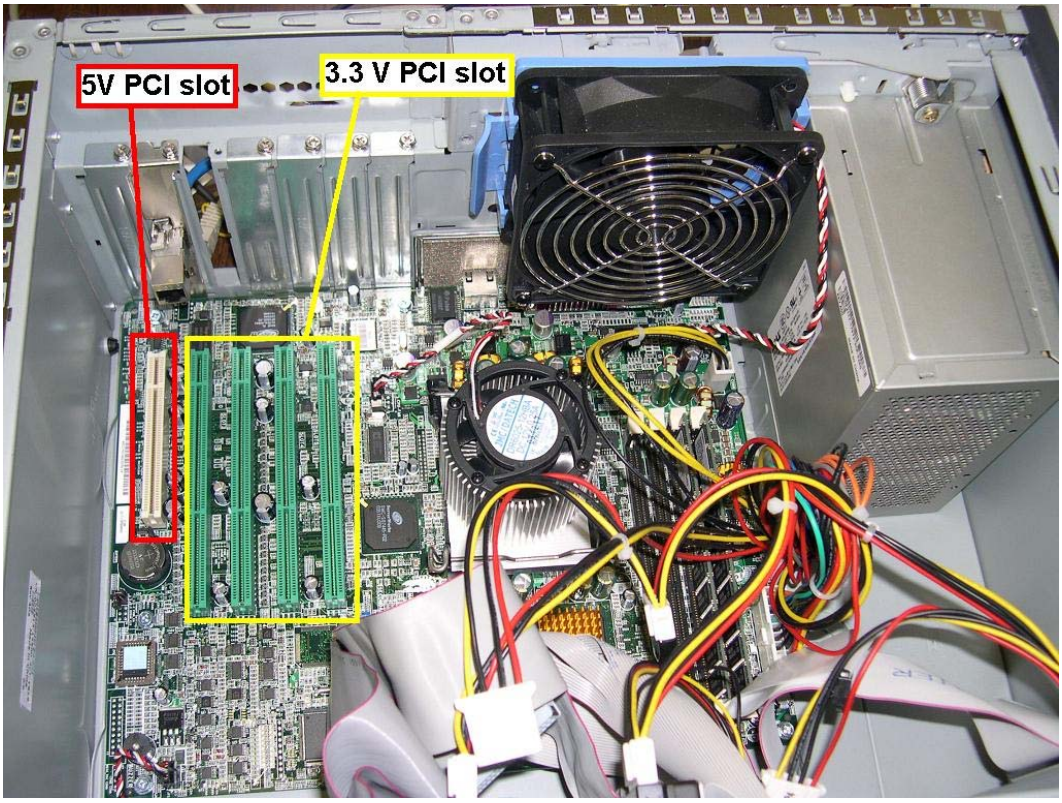

Figure 2-5 PCI Slots Highlighted Above. Note the longer length of 3.3V PCI slots compared to legacy 5V PCI slots

 Copyright © Signalogic 2006-2007

2) Be sure the SigC5561 card is inserted vertically into a 3.3V PCI slot.
3) Gently press down on the card until it settles into place and the card is fully seated into the 3.3V PCI slot, as shown in Figure 2-6 below. The backplate of the card should align flush with the back wall of the PC.
4) Screw in the mounting backplate bracket for stability and good grounding connection to the system chassis.

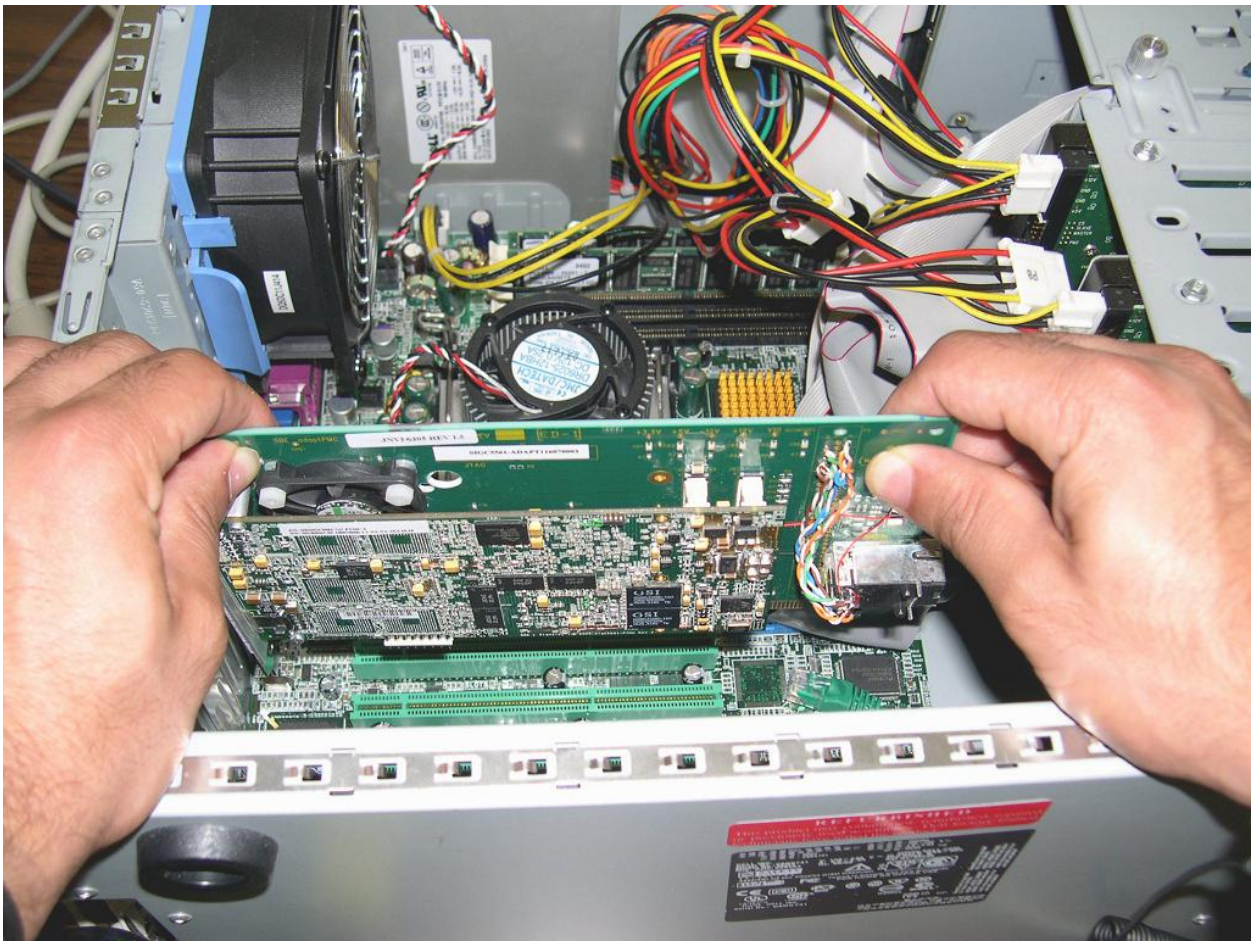

Figure 2-6 *Installing SigC5561 PCI Card*

Note: Always install the PCI card mounting screw, for both mechanical stability purposes and chassis grounding purposes.

Note: The SigC5561 card consumes a high amount of power (up to 23 W with all DSP devices installed), which makes it very important the card is fully seated into its PCI slot and the mounting bracket is screwed down. If the card is somehow knocked ajar, or otherwise not fully seated, it's conceivable that the loss of full metal connector contact on the power pins in the PCI slot could cause localized heat damage to either the motherboard or SigC5561 card.

2.2.6 Installing the SigC5561-PTMC Module

CAUTION Exercise extreme care when installing or removing the SigC5561-PTMC module to and from computer systems to prevent any possible damage. Please handle the module with care.

Note: First make sure power is completely off to the cPCI or ATCA carrier board (blade). The SigC5561 is not a hot-swap product! There should be no LED indicators of any type glowing on the carrier board. It's advisable to disconnect all AC power plugs from the cPCI or ATCA chassis. If you plug the SigC5561 module into a PMC or PTMC site with power still enabled, you will damage the module and void its warranty!

A cCPI carrier board with PTMC sites highlighted is shown in Figure 2-7 below.

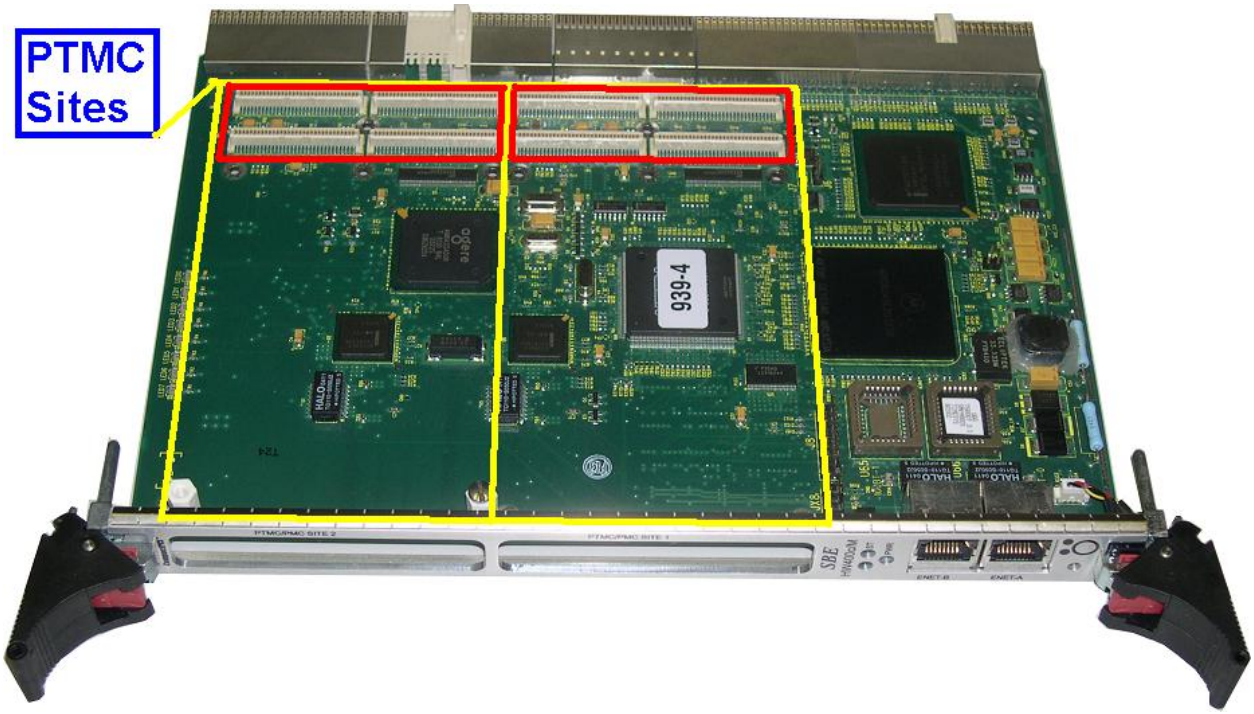

Figure 2-7 *cPCI Carrier Board with PTMC Sites Highlighted*

To install the SigC5561-PTMC module on a cPCI or ATCA carrier board, follow the steps below:

1) Insert the SigC5561-PTMC module into a 3.3V PTMC site on the cPCI or ATCA carrier board. A cPCI board with two PTMC sites is shown in Figure 2-7 above.
2) Gently press down on the module until it settles into place and the card is fully seated into the PTMC site, as shown in Figure 2-8 below.
3) Be sure the module is inserted completely flat in the PTMC site, and all four (4) PTMC connectors are fully seated.

4) Insert metal mounting screws for stability and system chassis ground connection. Mounting screws should have small circular head and not touch any PCB components.

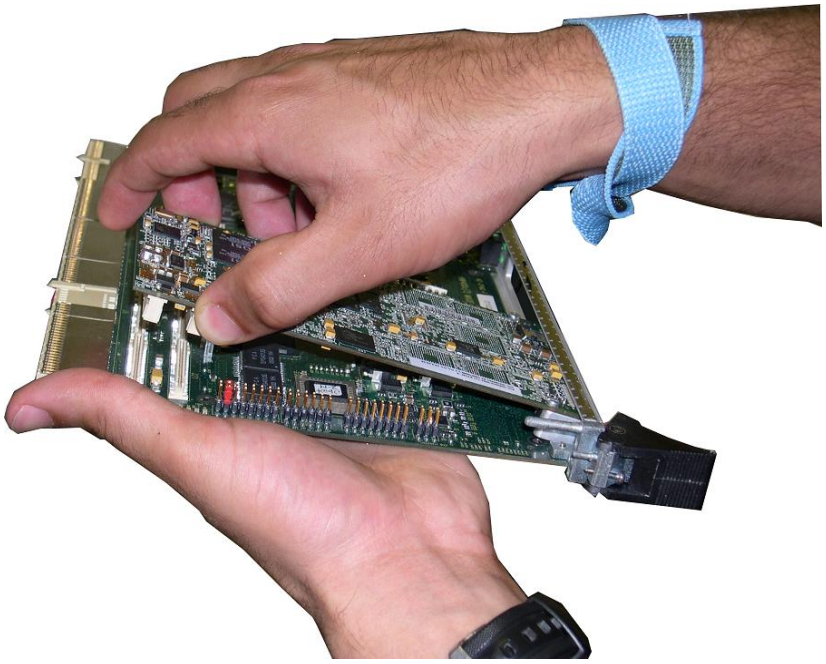

Figure 2-8 *Installing SigC5561 PTMC Module*

A cPCI carrier board with a SigC5561-PTMC module installed is shown in Figure 2-9 below.

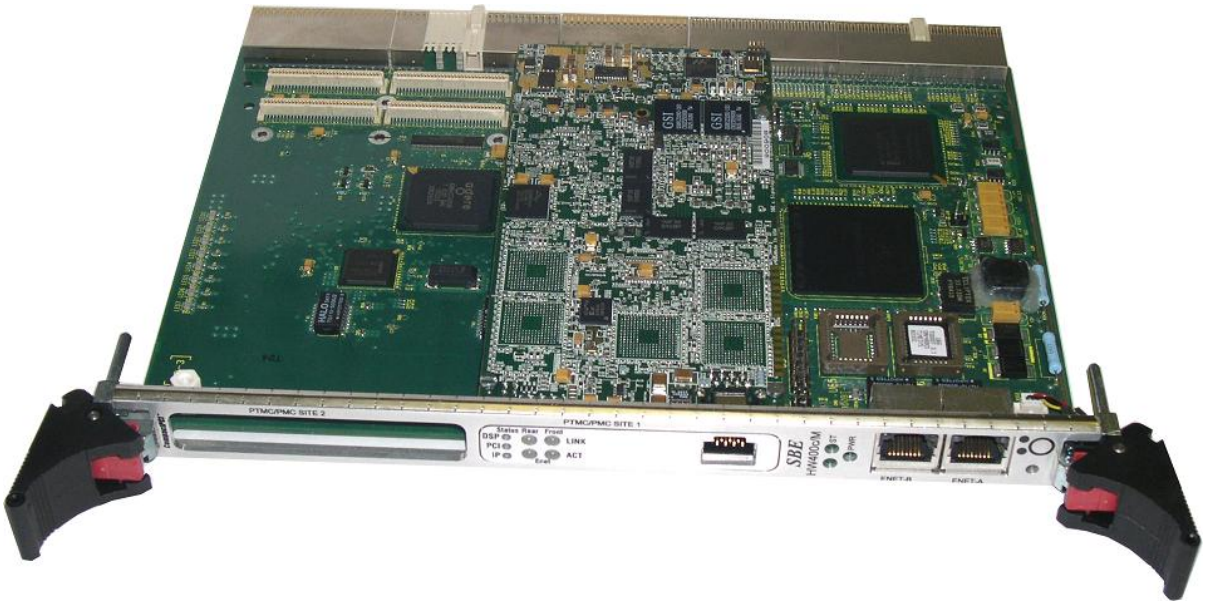

Figure 2-9 *cPCI Carrier Board with SigC5561-PTMC Module Installed*

 Copyright © Signalogic 2006-2007

Final Safety Check Before Power-On

After the PTMC module has been installed, perform the following crucial last-minute checks before applying power:

- Ensure PTMC connector pins are correctly inserted!! Look very closely. There should be no pins off by one row, or outside the connector
- Make sure the module is well-seated and pushed down firmly. The module should not be angled, or rock back and forth.

Note: It must be emphasized that if you plug the PTMC module in wrong it will likely be damaged due to a hideous and brutal condition, such as +12V going to a 3.3V IC pin, that will kill a logic device or other IC responsible for bus-interface on the card. In that case, necessary repairs are not covered under warranty!! So please be very careful and pay attention to detail during card installation.

2.2.7 GbE and 10/100 Ports

The SigC5561 card provides a GbE port for media (voice and data) traffic and a 10/100 port for control functions (FTP file download, remote API, test and debug). The card's default configuration loads a large code file at power-on boot time over the 10/100 port from an FTP server.

The 10/100 port is located on the backplate of the SigC5561 card, as shown in Figure 2-10 below.

The GbE port is located either on the backplate or internally on the card. In Figure 2-10 below, the GbE port is brought out to the backplate of an adjacent slot using a short internal CAT6 coupling cable.

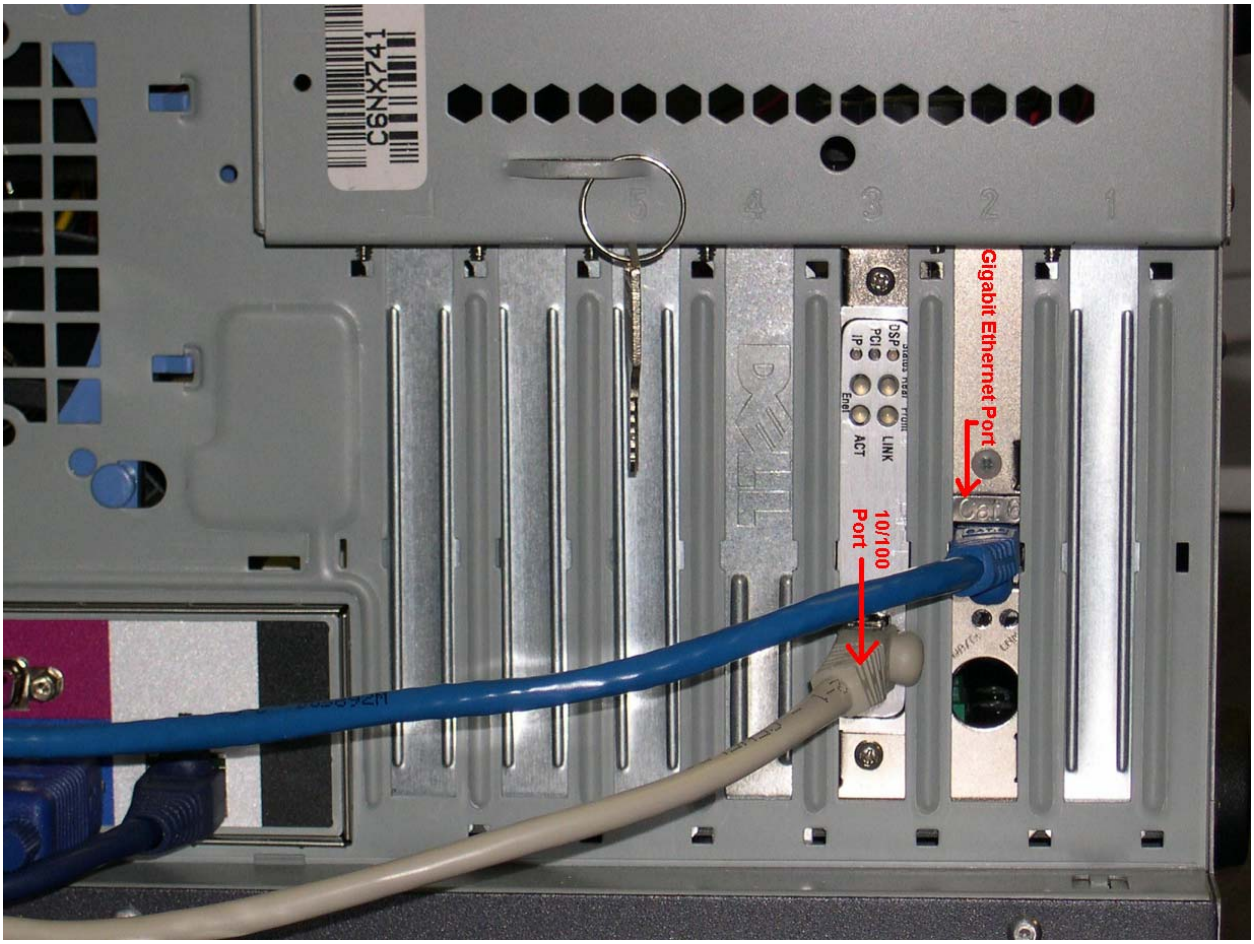

Figure 2-10 *Example of 10/100 and GbE Ports on Desktop PC Backplates*

In Figure 2-10 above, the RJ-45 connector visible on the extra backplate (connected to blue cable in the picture) is just a "coupler", or female-to-female adapter. If you need a part number for this adapter, please contact Signalogic.

2.2.8 Connecting Cables

Prior to SigC5561 power-on and boot, the 10/100 port should be connected to a network resource providing access to an FTP server. The 10/100 port is located on the backplate of the SigC5561 card (or front-panel bezel of the SigC5561-PTMC module), as shown in Figure 2-10 above.

For initial board bring up, the GbE port doesn't need to connect to anything. At some point prior to functional testing, the GbE port should be connected to a network resource that provides access to IP media (voice or data) RTP packets. As shown in Figure 2-10 above, the GbE port is brought out to the backplate of an adjacent slot using a short internal CAT6 coupling cable. This may not be possible in a 1U server configuration or a server with no extra backplates. In that case, the GbE cable can be connected to the internal connector and the cable routed out of the PC enclosure in some way that is suitable for the application.

The 10/100 port may be connected using either CAT5 or CAT6 cable and common variety 10/100 switches and hubs.

The GbE port may only be connected using CAT6 cable and GbE capable switches.

2.2.9 FTP Server

Before applying power to the SigC5561 product, an FTP server must be running to provide files needed by the WinPath network processor during its power-on boot phase. Please follow the instructions in section 3.2.2, FTP Server Setup, then return to section 2.3, Power-On, to proceed.

2.3 Power-On

After the SigC5561 PCI card or SigC5561-PTMC module is correctly installed in the computer system and the 10/100 port is connected as explained in section 2.2.6 above, the SigC5561is now ready to be powered on to verify initial boot.

2.3.1 Panel LED Indicators at Power-On

Note: the following panel LED description applies to both the SigC5561 PCI card and SigC5561-PTMC module products. For the PCI card, the panel LEDs appear on the backplate of the card; for the PTMC module the LEDs appear on the front bezel of the module.

Upon power on, the following panel LED indicators should be checked to verify a successful board boot.

Panel LED Label	**Meaning**	**Acceptable Functionality, Depending on Hardware Revision**
PCI Status	Green or blinking green (unless the SigC5561 is booted in “stand-alone” mode; see section 5.2.5)	Off
DSP Status	Off at power-on, green after DSP download, blinking green when voice channels are active, red if DSP error occurs	Always red
IP Status	Green or blinking green if RTP packet data is active	Off
Rear Enet LINK	Green if GbE port is connected and link is established	Red at power-on, turns green after DSP download
Rear Enet ACT	Blink red and green if GbE port is connected, blink green if link has traffic	Same
Front Enet LINK	Green if 10/100 port is connected and link is established	Always Red
Front Enet ACT	Blink red and green if GbE port is connected, blink green if the link has traffic	Always Red

Note that a visual description of the panel LED boot pattern is given in Figure 2-11 below.

Note: If no LEDs are lit, then turn off (shutdown) your system immediately, disconnect the AC power plugs, and remove the card.

If panel LEDs are on but the boot pattern does not match the above description and image shown in Figure 2-11 below, then refer to section 4.4, Troubleshooting.

 Copyright © Signalogic 2006-2007

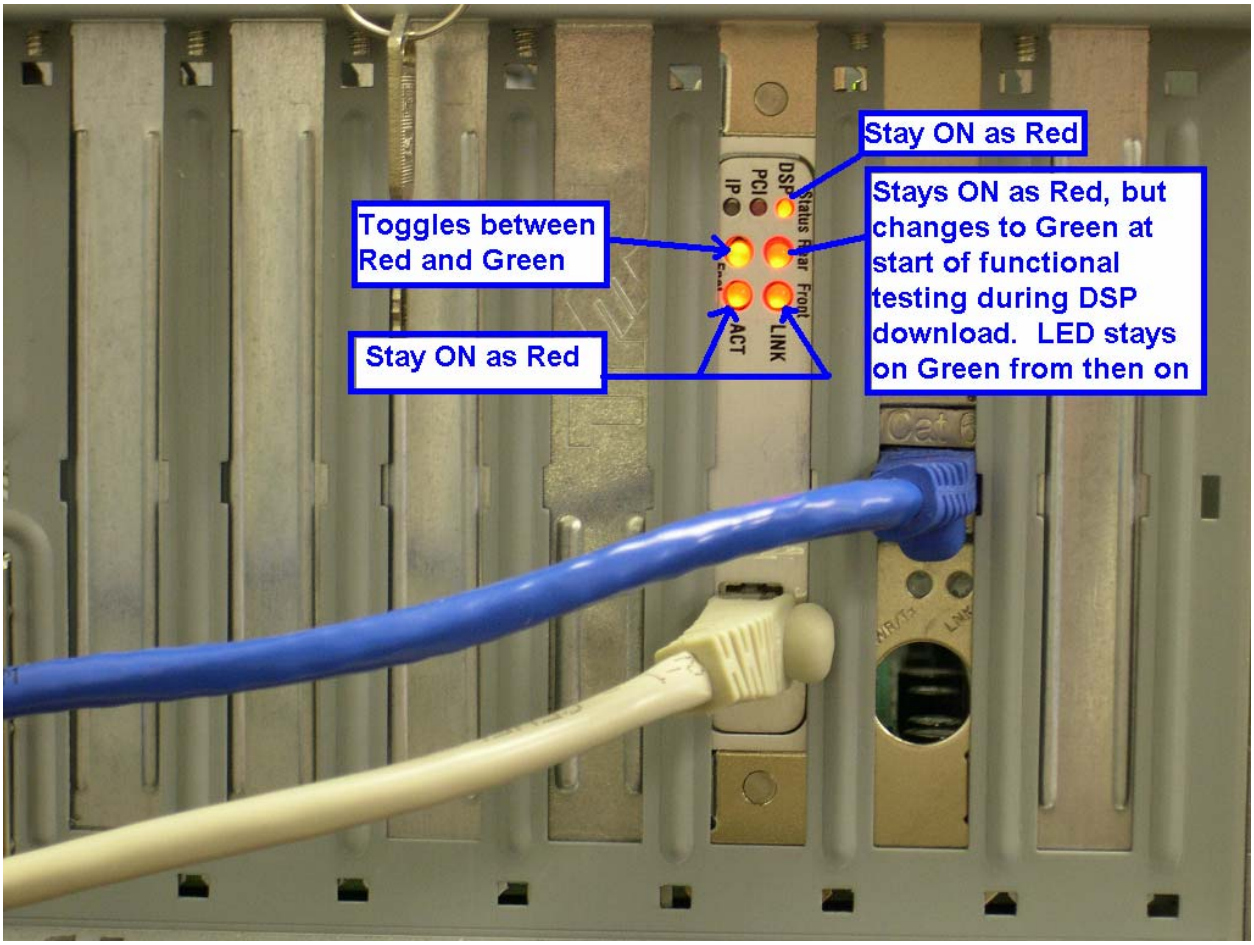
Stay ON as Red
Stays ON as Red, but changes to Green at start of functional testing during DSP download. LED stays on Green from then on
Toggles between Red and Green
Stay ON as Red
DSP
PCI
IP
Status Rear Front
ACT
LINK

Figure 2-11 *Panel LED Indicator Pattern at Power On*

For more information about panel LED indicators, see section 7.6.1, Panel LED Indicators.

 Copyright © Signalogic 2006-2007

2.3.2 Status / Error Code Indicators at Power-On

Note: the following status / error code LED description applies to both the SigC5561 PCI card and SigC5561-PTMC module products. The status / error code LEDs are a row of 16 small "surface mount" LEDs located on the top side of the SigC5561 card or module, near the panel LEDs. See Figures 2-1 and 2-2 above for a graphical depiction of location of the status / error code LEDs.

Upon power on, the following status / error code LED indicators should be checked to verify a successful board boot.

Status / Error Code LED Number	Meaning	Acceptable Functionality, Depending on Hardware Revision
1	On = Logic boot Ok Off = Logic boot failed	
2	ND [1]	
3	ND	
4	ND	
5	ND	
6	ND	
7	ND	
8	On = PCI bus detected Off = stand-alone mode	
9	On = UTOPIA bus read activity	Off
10	On = UTOPIA bus write activity	Off
11	BIST [2] Code 0 = No errors, status Ok 1 = Main logic clock error 2 = TDM Reference clock error 3 = UTOPIA clock error 4 = WinPath boot error	All Off
12		
13		
14		
15		
16	Blinking indicates logic 'heartbeat' is normal, logic continues to function normally	

Notes

[1] ND = Not Defined
[2] BIST = Built-In Self-Test

Note that a visual description of acceptable status / error code LED boot pattern is given in Figure 2-12 below.

Note: If no LEDs are lit, then turn off (shutdown) your system immediately, disconnect the AC power plugs, and remove the card.

If status / error code LEDs are on but the boot pattern does not match the above description and image shown in Figure 2-12 below, then refer to section 4.4, Troubleshooting.

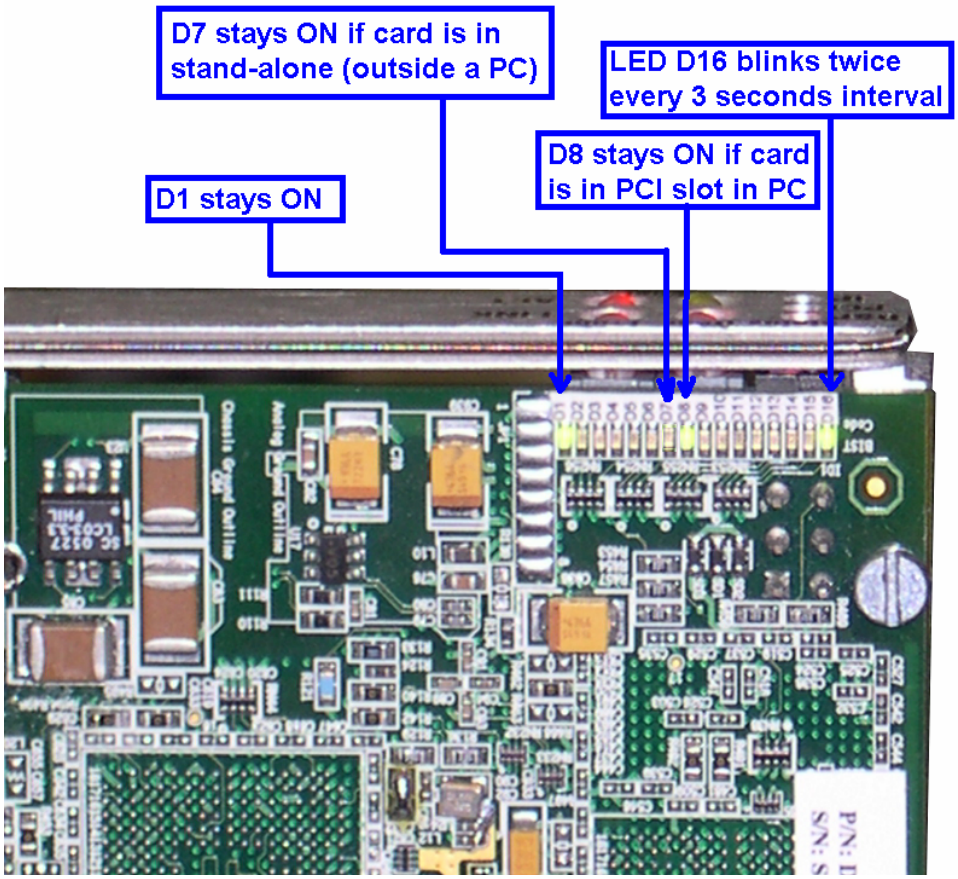
D7 stays ON if card is in stand-alone (outside a PC)
LED D16 blinks twice every 3 seconds interval
D8 stays ON if card is in PCI slot in PC
D1 stays ON

Figure 2-12 *Status / Error Code LED Indicator Pattern at Power On*

For more information about status / error code LED indicators, see section 7.6.2, Status / Error Code LED Indicators.

2.3.3 Verifying PCI Status

After the SigC5561 card has been powered on and the LED visual pattern described in section 2.3.1 above has been verified, log in to your Linux system as root or sudo-root and open a command terminal in the GUI environment of Linux. As an example, to open a GUI terminal in CentOs, click on:

```
Applications -> System Tools -> Terminal
```

Once the terminal window is open, enter the following command to verify the SigC5561 and necessary PCI devices in the system are recognized and supported by the Linux kernel:

```
lspci
```

You should see a display similar to Figure 2-13 below, listing Ethernet controller, PCI Bridge (multiple devices), and the SigC5561 card as a co-processor device.

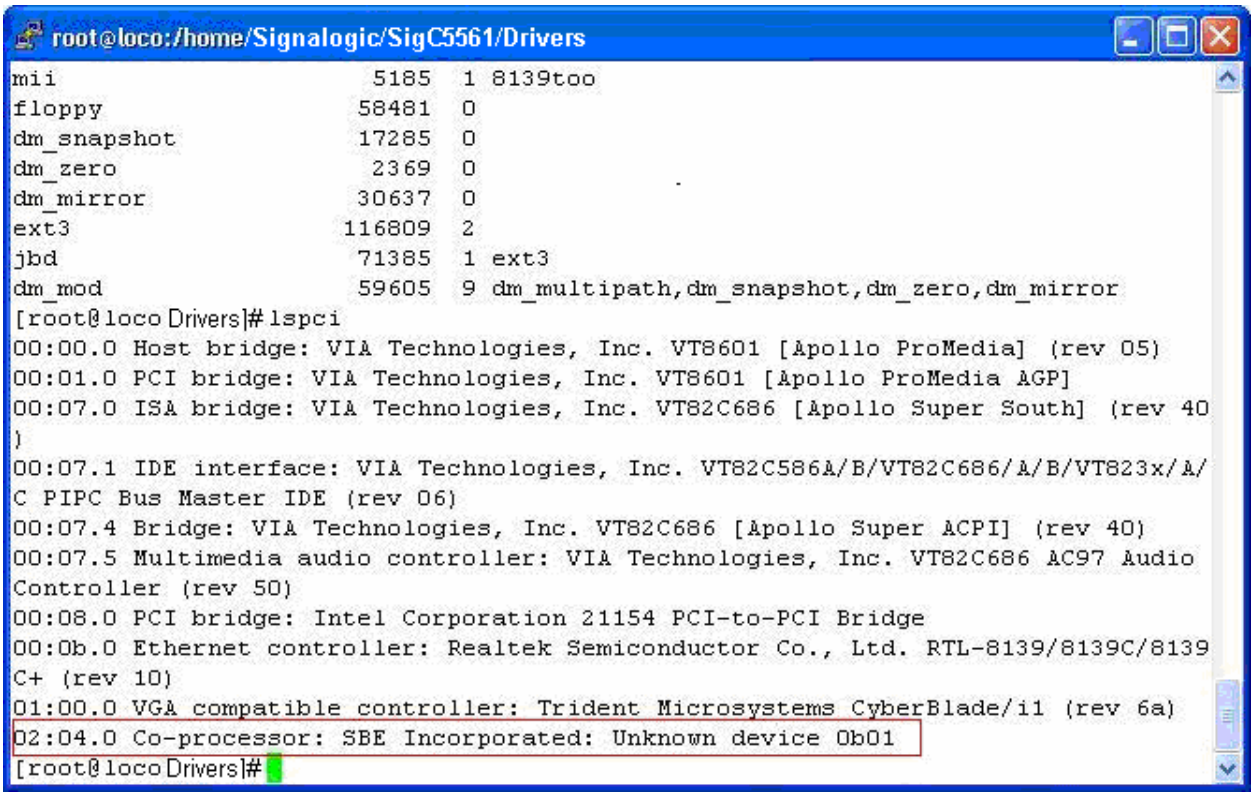
root@loco:/home/Signalogic/SigC5561/Drivers
mii 5185 1 8139too
floppy 58481 0
dm_snapshot 17285 0
dm_zero 2369 0
dm_mirror 30637 0
ext3 116809 2
jbd 71385 1 ext3
dm_mod 59605 9 dm_multipath,dm_snapshot,dm_zero,dm_mirror
[root@loco Drivers]# lspci
00:00.0 Host bridge: VIA Technologies, Inc. VT8601 [Apollo ProMedia] (rev 05)
00:01.0 PCI bridge: VIA Technologies, Inc. VT8601 [Apollo ProMedia AGP]
00:07.0 ISA bridge: VIA Technologies, Inc. VT82C686 [Apollo Super South] (rev 40
)
00:07.1 IDE interface: VIA Technologies, Inc. VT82C586A/B/VT82C686/A/B/VT823x/A/
C PIPC Bus Master IDE (rev 06)
00:07.4 Bridge: VIA Technologies, Inc. VT82C686 [Apollo Super ACPI] (rev 40)
00:07.5 Multimedia audio controller: VIA Technologies, Inc. VT82C686 AC97 Audio
Controller (rev 50)
00:08.0 PCI bridge: Intel Corporation 21154 PCI-to-PCI Bridge
00:0b.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139
C+ (rev 10)
01:00.0 VGA compatible controller: Trident Microsystems CyberBlade/i1 (rev 6a)
02:04.0 Co-processor: SBE Incorporated: Unknown device 0b01
[root@loco Drivers]#

Figure 2-13 *'lspci' Command Result*

If you see messages such as "unknown device", then take the following steps:

At Linux command prompt, type in command:

`cd /usr/share/hwdata`

and press the <Enter> key, then type in the command:

`vi pci.ids`

and press <Enter> again. Commands are shown in Figure 2-14 below.

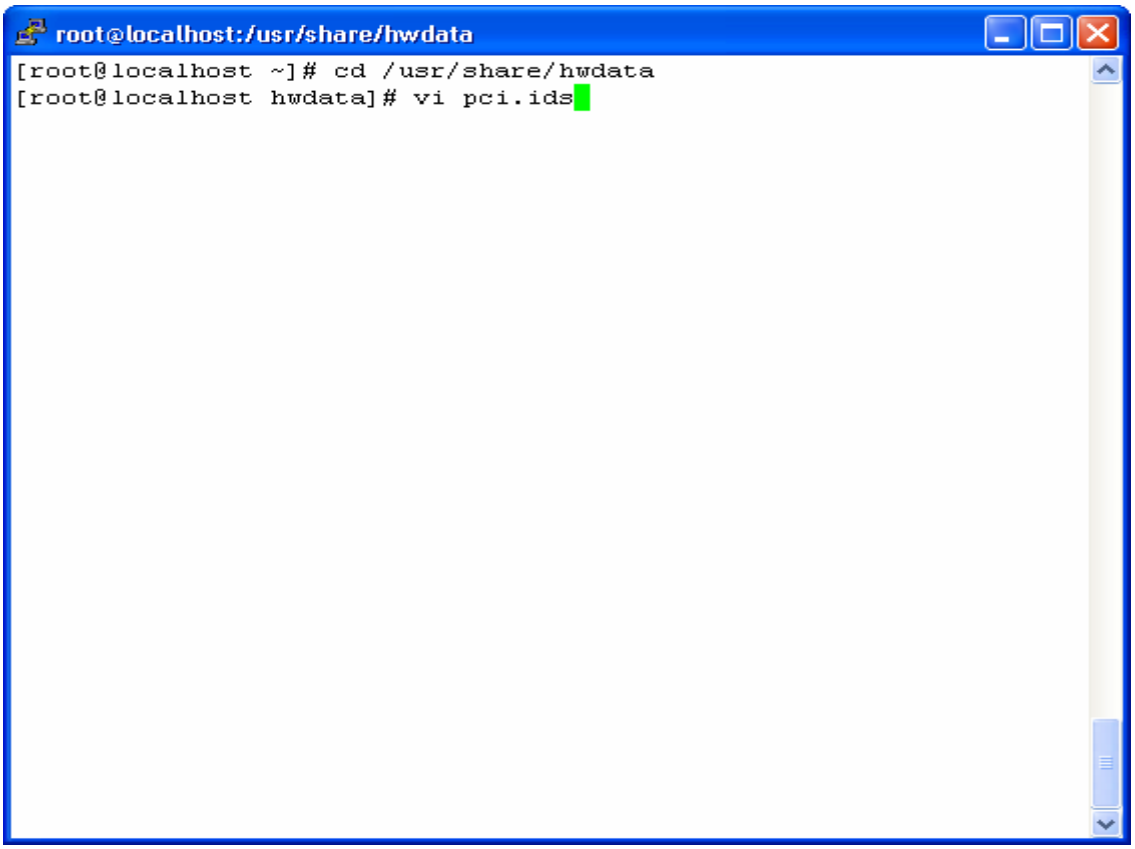

Figure 2-14 *Editing the 'pci.ids' File*

Then, within the 'pci.ids' file, add the line

```
0b01  SigC5561 VoIP card
```

Make sure to use the <Tab> key to insert a tab to move the **0b01** text over from the left margin (don't use spaces), and make sure the column alignment for this line matches with the existing lines already in the file. Refer to Figure 2-15 below.

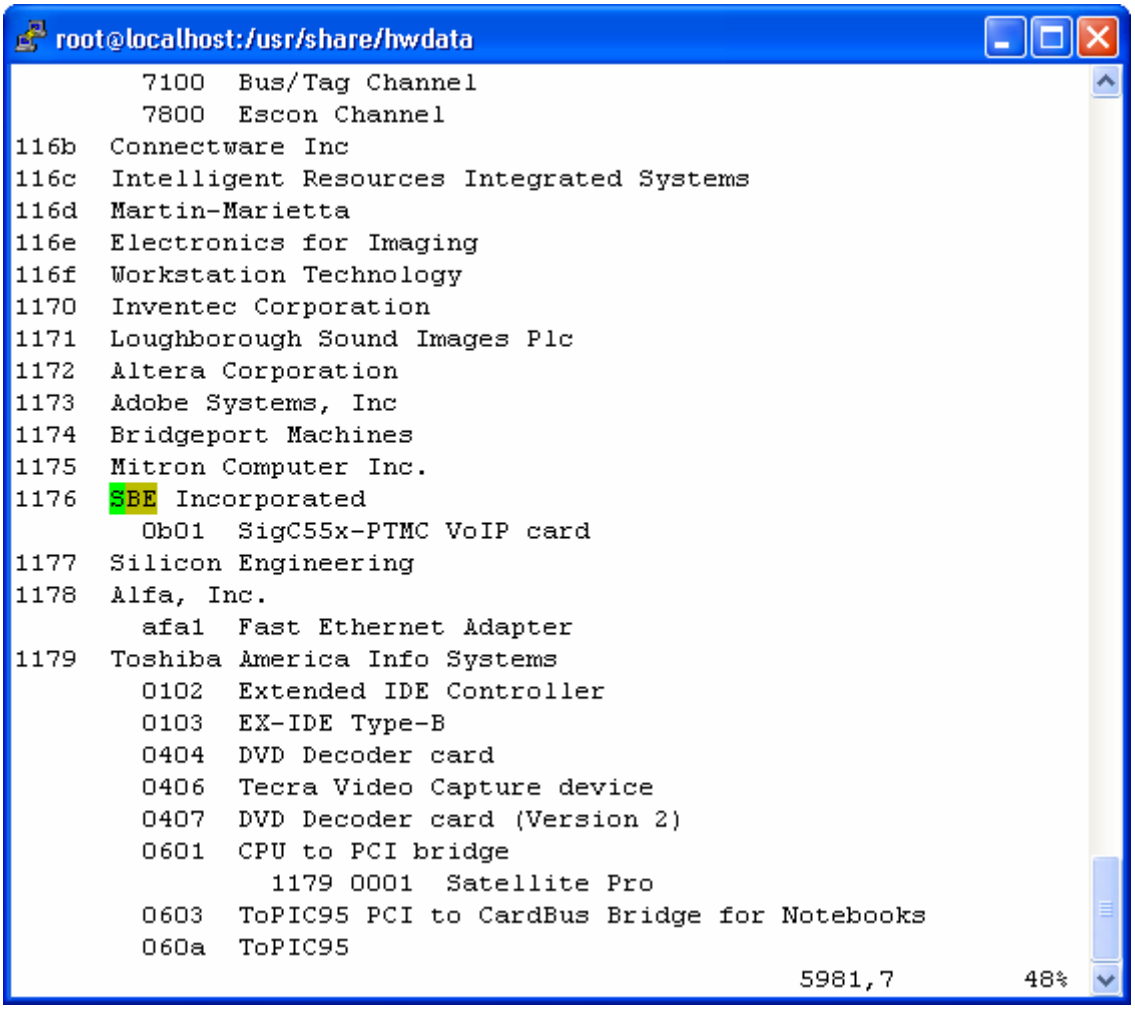

Figure 2-15 *Correct Alignment of Device IDs in 'pci.ids' File*

If you still see "unknown device" displayed for the SigC5561 card after doing another 'lspci' command, please contact Signalogic technical support for additional assistance.

Note: If you see unrecognized PCI bridge or other motherboard devices, then the SigC5561 card may still operate correctly. And, it may be possible to edit the pci.ids file to avoid "unknown device" labeling for such devices, using the method described above. However, please keep in mind that this method is a "cosmetic" work-around that may cover up deficiencies due to not actually installing the latest driver. The best approach to ensure correct operation for any PCI device is to make sure the latest driver is installed.

This completes the SigC5561 PCI card installation. Please proceed to sections 3 and 4 below, Software Installation and Test and Checkout.

 Copyright © Signalogic 2006-2007

3 Software Installation

Note: If a printed instruction sheet has been included with the Signalogic software CD, please follow that first, then proceed to specific steps below.

During software installation, it may be helpful to consult the following links depending on specific type of software or hardware.

http://www.signalogic.com/tech_support.htm#Getting_Started

ftp://ftp.signalogic.com/documentation/Boards/SigC67xx/SigC67xxSetup.PDF

ftp://ftp.signalogic.com/documentation/Boards/SigC67xx/VDS_Users_Guide.pdf

ftp://ftp.signalogic.com/documentation/DirectDSP/Users Guide/

ftp://ftp.signalogic.com/documentation/DirectDSP/Reference Guide/

ftp://ftp.signalogic.com/documentation/SCI/C5xC6xCCS JTAG.PDF

The above links are intended only as supplemental information… please use the sections below as the main guide for SigC5561 software installation.

3.1 Software Checklist

The following sections cover Signalogic-provided software and third-party software recommended for functional test and checkout.

3.1.1 Signalogic Software Checklist

The following software items are provided by Signalogic either on CD enclosed with the SigC5561 product shipment or downloadable via Signalogic secure FTP site. FTP site download is the recommended way to update SigC5561 software, which may change frequently. Secure FTP login information is provided on customer-by-customer basis.

Signalogic-provided software for the SigC5561 product is listed below:

1. VxWorks files (including but not limited to "startup.script", "dimt_boot.txt", and VxWorks executable). These files are downloaded via FTP server by the SigC5561 card or module at boot-time.
2. Linux driver. Tested on Red Hat Release 8.0, kernel version 2.4.20-8; and CentOS Release 4.4, kernel version 2.6.9-42.ELsmp
3. DirectDSP software. DSP hardware product API.
4. DSPower-MGTS software. VoP and TDM interface API.
5. Utilities, including Flash program, PCI terminal emulation program, HPI test program, etc.
6. Test programs, including C source code and make files that use the DirectDSP and DSPower-MGTS APIs.

3.1.2 Functional Test Software Checklist

Third-party software recommended for SigC5561 product functional test and checkout is listed below:

1. FTP server software. WFTPD (Windows FTP Daemon) is recommended by Signalogic although other FTP server utilities should work just as well. WFTPD can be downloaded at http://www.wftpd.com/wftpdreg.htm. For normal desktop running Windows 2000, XP, 2003, or Vista, WFTPD Server 3.25 (standard version) is recommended (http://www.wftpd.com/downloads/wftpd325.zip). For Windows Servers running Windows Server 2000 or 2003, WFTPD Pro Server 3.25 is recommended (http://www.wftpd.com/downloads/protr325.zip).
2. Ethereal is recommended for packet monitoring test verification. Ethereal is available for download at web site http://www.ethereal.com/download.html.

3.1.3 Directory Structure

Below is an overview of what directory structure should like after successful installation of SigC5561 software.

Linux Host Software Directory Structure

```
/home/Signalogic
   |
   |__DirectDSP
   |       |
   |       |__HwLib
   |       |
   |       |__VoP
   |       |
   |       |__TDM
   |
   |
   |__SigC5561
           |
           |__Drivers
                 |
           |     |__Makefiles
           |           |
           |           |__CentOS_26
           |           |
           |           |__RedHat_24
           |
           |__Test_Programs
           |         |
           |         |__HPI_PCI
           |         |
           |         |__terminal
           |         |
           |         |__testlib
           |
           |__Utilities
                 |
                 |__flash

/etc/Signalogic
     |
     |_board.conf
```

3.1.4 Electronic Format Software Upgrade

For login information to the secure FTP site, please contact Signalogic. If you are downloading SigC5561 software via secure FTP site or receiving a .zip or .gz file by e-mail, the following procedures are typical:

1) Before attempting to rebuild, unzip or untar files to correct directory locations, compare the name of the compressed file to the directory structure in section 3.1.3 above in order

to determine which subfolders are being updated. ***If you have previously modified any Signalogic-provided files, then be aware they may be overwritten by the unzip or untar procedure!*** Files that you have created, or Signalogic-provided files that you have renamed, should be safe from being overwritten.

2) On the Signalogic's secure FTP site, the current SigC5561 software file is named as "Signalogic_distro_***date***.zip", where ***date*** is the updated date for software. Unzip the software file, which contains the README file, the *.tar.gz file (follow step 3 below for installing the tar.gz file), and the 'board.conf' file. Please create a folder name /etc/Signalogic on your Linux machine and copy the 'board.conf' file to that location. Other files on secure FTP site need to be copied to your Windows machine with the FTP server running. Refer to section 3.1.2 and 3.2.2 for FTP Server setup on your Windows machine. The files 'dimt_boot.txt' and 'startup.script' should be copied to directory C:\test, and file 'vxWorks' should be copied to directory C:\tftp\VxWorks.

3) To untar (uncompress) the SigC5561 software into your Linux machine, go the directory /home, and execute command 'tar –xzf Signalogic_distro.tar.gz'. After untarring, verify the directory structure created as showed above.

4) Files in the .gz or .zip file may be read-only (due to version control software); if so you may need to reset file attributes before using or add to version control software as needed.

3.1.5 Supported Linux OS Installation

The following Linux Operating Systems have been tested and are supported by Signalogic:

1. **CentOS**

   CentOS version 4.4, with kernel version 2.6.9, .iso images can be downloaded at http://centos.secsup.org/4.4/isos/i386/. Please download and install all of the following four (4) images in order to have a full server version of CentOS:

   a. CentOS-4.4-i386-bin1of4.iso

   b. CentOS-4.4-i386-bin2of4.iso

   c. CentOS-4.4-i386-bin3of4.iso

   d. CentOS-4.4-i386-bin4of4.iso

   After successful installation of CentOS, be sure there is a listing of kernel source code in directory /usr/src/kernels/. If not, please download the kernel source code with kernel version 2.6.9 into the directory /usr/src/kernels/.

2. **Red Hat**

   Red Hat version 8, with kernel version 2.4.18, .iso images can be downloaded at http://www.redhat.com/.

   After successful installation of Red Hat, be sure there is a listing of kernel source code in directory /usr/src/. If not, please download the kernel source code with kernel version 2.4.18 into the directory /usr/src/.

3.2 Software Setup and Configuration

The following software installations are necessary to perform SigC5561 product functionality.

3.2.1 Board Configuration File

A "board.conf" file must exist in the following folder:

```
/etc/Signalogic
```

A default board.conf file is installed by standard software installation, but it's imperative that you manually check the board.conf file to make sure it matches your SigC5561 product! See section 3.1.3 above, Directory Structure, for more detail on folders used by SigC5561 software.

Note: if the board.conf file does match the physical configuration of the SigC5561 PCI card or SigC5561 PTMC module, then software test and checkout results may be undefined.

For a SigC5561 product with 2-DSP configuration, the board.conf file should look as follows:

```
[SigC5561]

num_dsp_module=2
winpath_clock=166
```

For a 7 DSP configuration, the board.conf file should look as follows:

```
[SigC5561]

num_dsp_module=7
winpath_clock=166
```

If you have more than one SigC5561 product with different DSP configurations, use the smaller value in the board configuration file. This limitation applies to test and checkout utilities, but not to general application layer programs. When building application-specific programs, the DSAssignCard() function allows DSP configuration to be specified on card-by-card basis.

3.2.2 FTP Server Setup

A number of FTP server utilities are widely available online. Signalogic lab testing typically uses the "Standard" version of the WFTPD utility (see section 3.1.2 above), installed and running on a separate WinXP machine connected to the same network as the SigC5561 product. The 'About Box' for the WFTPD software used at Signalogic is shown in Figure 3-1 below:

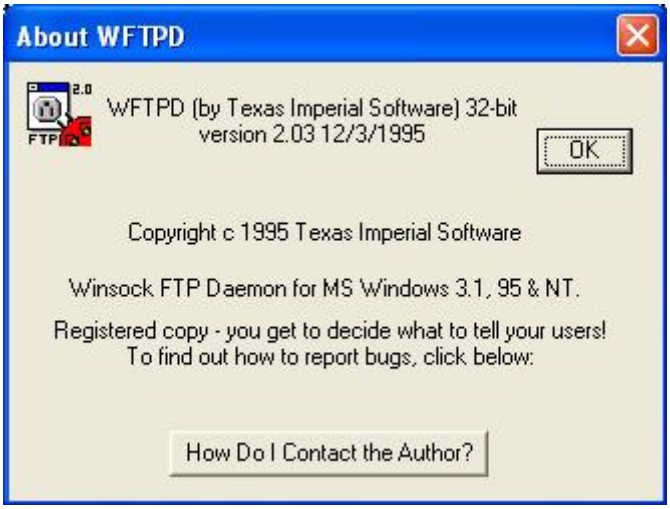
About WFTPD
WFTPD (by Texas Imperial Software) 32-bit
version 2.03 12/3/1995
OK
Copyright c 1995 Texas Imperial Software
Winsock FTP Daemon for MS Windows 3.1, 95 & NT.
Registered copy - you get to decide what to tell your users!
To find out how to report bugs, click below:
How Do I Contact the Author?

Figure 3-1 *WFTPD Software About Box*

A “Professional” version of WFTPD is also available, but not required unless the host machine for the FTP server is running *Windows Server 2000* or *Windows Server 2003* OS, in which case WFTPD Pro Server 3.25 is recommended (see section 3.1.2 above, Functional Test Software Checklist, for a download URL).

An alternative solution is to use a Linux FTP server, running on either a machine separate from the SigC5561 product or the same machine. However, for first-time functional test and checkout, ***a separate machine is strongly recommended***, in case there should be PCI, hardware driver, or other low-level machine / configuration issues during boot. In such a situation, a separate machine helps to minimize confusion as to what is actually going happening.

Following is a list of steps to configure the WFTPD utility:

1. Run the WFTPD utility on a WinXP machine and select ‘Security’ main menu option, as shown in Figure 3-2 below. Make the following settings:

Computer Name	jumjumbo
IP address	10.0.0.29

2. Set up a user with

User name	yjye
Password	yjye

Note the above settings are default values assumed by the WinPath network processor on the SigC5561 product at boot-time. To change the default values, a utility called “ftpconfig” is provided with the SigC5561 product software; see section 5.4, FTP Download, for more information.

Note also that it's possible to avoid FTP download altogether and place executable VxWorks code into Flash located on the SigC5561. For more information, see section 7.5.2, Network Processor Code Flash Configuration.

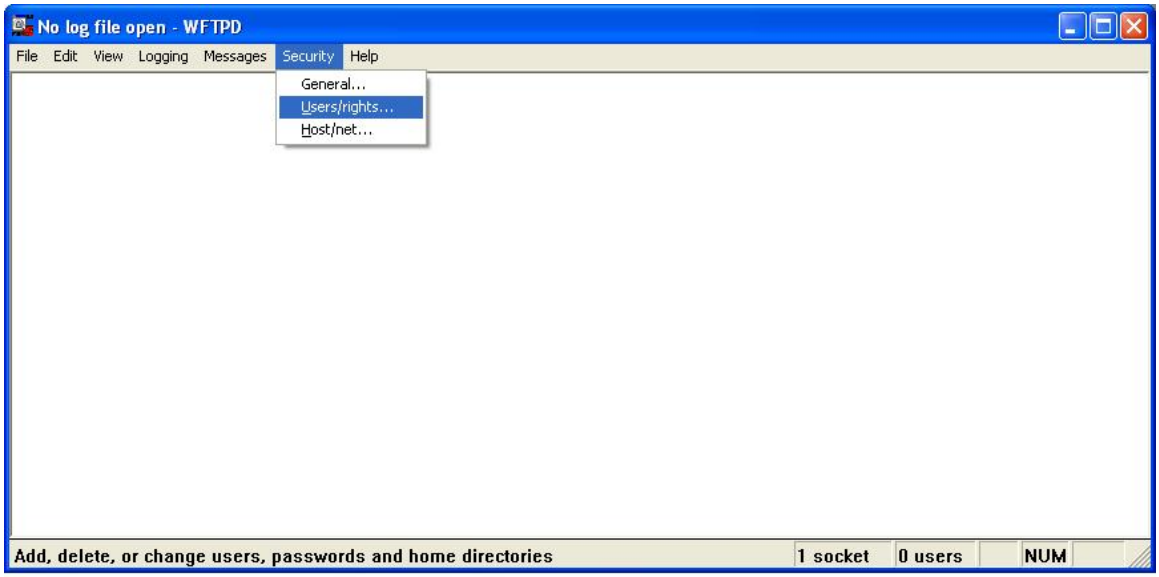

Figure 3-2 *Setting Up FTP User Account, Step 1*

3. The home directory field should be set to:

```
C:\tftp\VxWorks
```

Save the "vxworks" executable image (code) file in this directory. The SigC5561 card or module will download this code file via its 10/100 port at power-on boot.

Figure 3-3 *Setting Up FTP User Account, Step 2*

4. Create a folder:

```
C:\test\
```

on the FTP server machine and save the files “startup.script” and “dimt_boot.txt” in this folder. The SigC5561 card or module will download these text files via its 10/100 port at power-on boot.

3.2.3 SigC5561 Linux Driver

To build and load the Linux driver for the SigC5561 PCI card or PTMC module, follow the steps below.

1. Log in as root. This is required for loading a hardware driver module.
2. To make the driver module for your system, an appropriate Makefile is needed.
   a. For Red Hat,

      If Red Hat with kernel version 2.4.xx is installed on your system, then change the working directory to be:

      ```
      /home/Signalogic/SigC5561/Drivers/Makefiles/RedHat_24
      ```

      Do a ‘ls’ command to see the available Makefiles under the RedHat_24 directory. Copy the file ‘Makefile_2.4.18’ to the Drivers folder as file name ‘Makefile’ with command ‘cp Makefile_2.4.18 ../../Makefile’ as shown below. The kernel version in the file name indicates the supported Red Hat kernel version for the Makefile.

   b. For CentOS,

      If CentOS with kernel version 2.6.xx is installed on your system, then change the working directory to be:

      ```
      /home/Signalogic/SigC5561/Drivers/Makefiles/CentOS_26
      ```

      Do a ‘ls’ command to see the available Makefiles under the CentOS_26 directory. Copy the file ‘Makefile_2.6.9’ to the Drivers folder as file name ‘Makefile’ with command ‘cp Makefile_2.6.9 ../../Makefile’ as shown below. The kernel version in the file name indicates the supported CentOS kernel version for the Makefile.
3. At the Linux command prompt, change your current folder to be:

   ```
   /home/Signalogic/SigC5561/Drivers
   ```
4. Execute the following commands, press <Enter> key after each one

   ```
   make clean
   make
   make load
   ```
5. To verify if the module is loaded, enter the command:

   ```
   lsmod
   ```

   You should see a module named “sigc55x” in the list, as show in Figure 3-4 below.

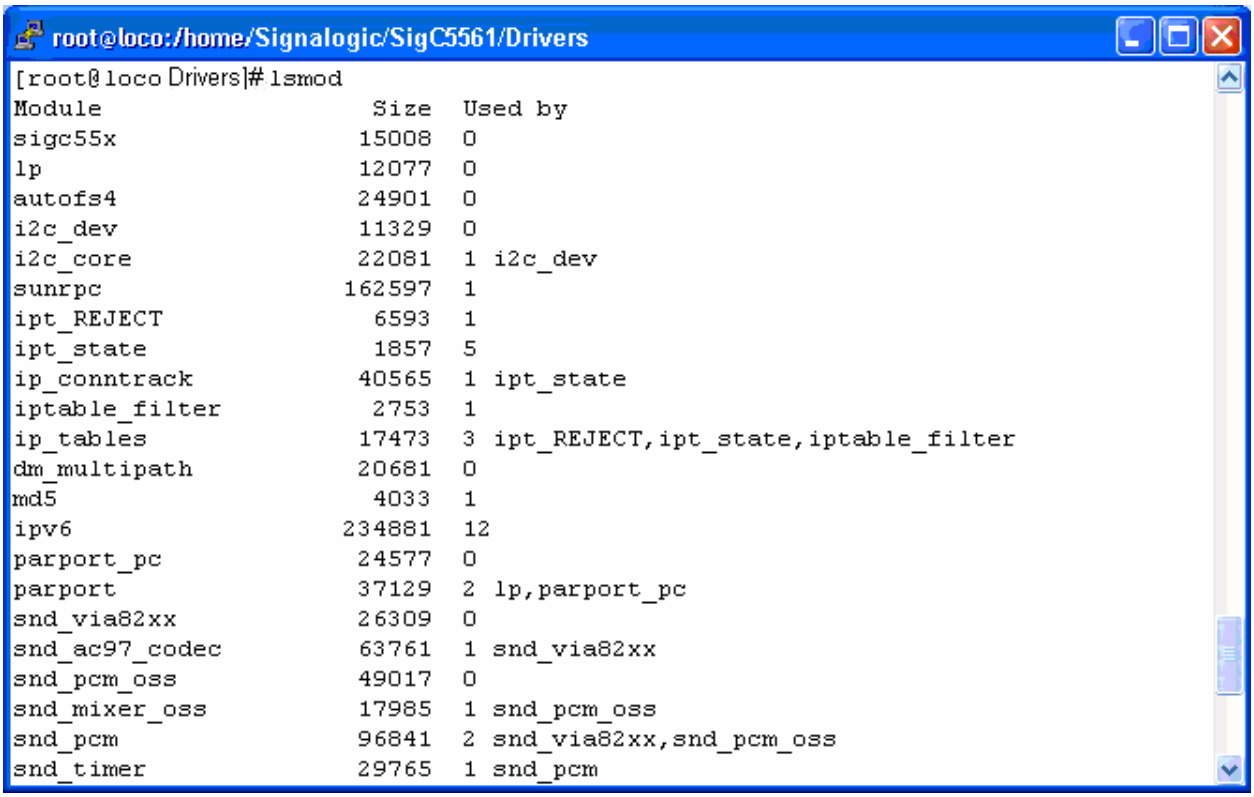

Figure 3-4 *Successful SigC5561 Card Driver Load*

If the Linux kernel version does not support the driver code provided by Signalogic, then you should see an error message so indicating; in that case please contact Signalogic tech support.

To unload (remove) the driver module, enter the command:

```
rmmod sigc55x
```

After removing, the driver module may be reloaded by entering the 'make load' command again from the /home/Signalogic/SigC5561/Drivers directory. Note that you must be logged in as root to unload the driver module.

3.2.4 Ethereal Installation

Ethereal is a software-based network protocol analyzer, downloadable over the Internet web site (http://www.ethereal.com/download.html).

After download, follow the installation instructions provided by Ethereal. Ethereal user documentation is also available at web site (http://www.ethereal.com/docs/).

The Signalogic functional tests described in the following sections combine Ethereal with a Port Mirroring Gigabit Switch to monitor voice data traffic to/from the SigC5561 product and verify voice codec, RTP, and other network packet parameters.

The 'About Box' for the Ethereal software used at Signalogic is shown in Figure 3-5 below:

About Ethereal
Ethereal
Authors
Folders
Plugins
Ethereal - Network Protocol Analyzer
Version 0.99.0
Copyright 1998-2006 Gerald Combs <gerald@ethereal.com> and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Compiled with GTK+ 2.6.9, with GLib 2.6.6, with WinPcap (version unknown),
with libz 1.2.3, with libpcre 6.4, with Net-SNMP 5.2.2, with ADNS, with Lua 5.1.
Running with WinPcap version 3.1 (packet.dll version 3, 1, 0, 27), based on libpcap version 0.9[.x] on Windows XP S, build 2600.
Ethereal is Open Source Software released under the GNU General Public License.
Check the man page and http://www.ethereal.com for more information.
OK

Figure 3-5 *Ethereal Software About Box*

4 Test and Checkout

Below is a diagram of the general test setup expected by the test and checkout programs:

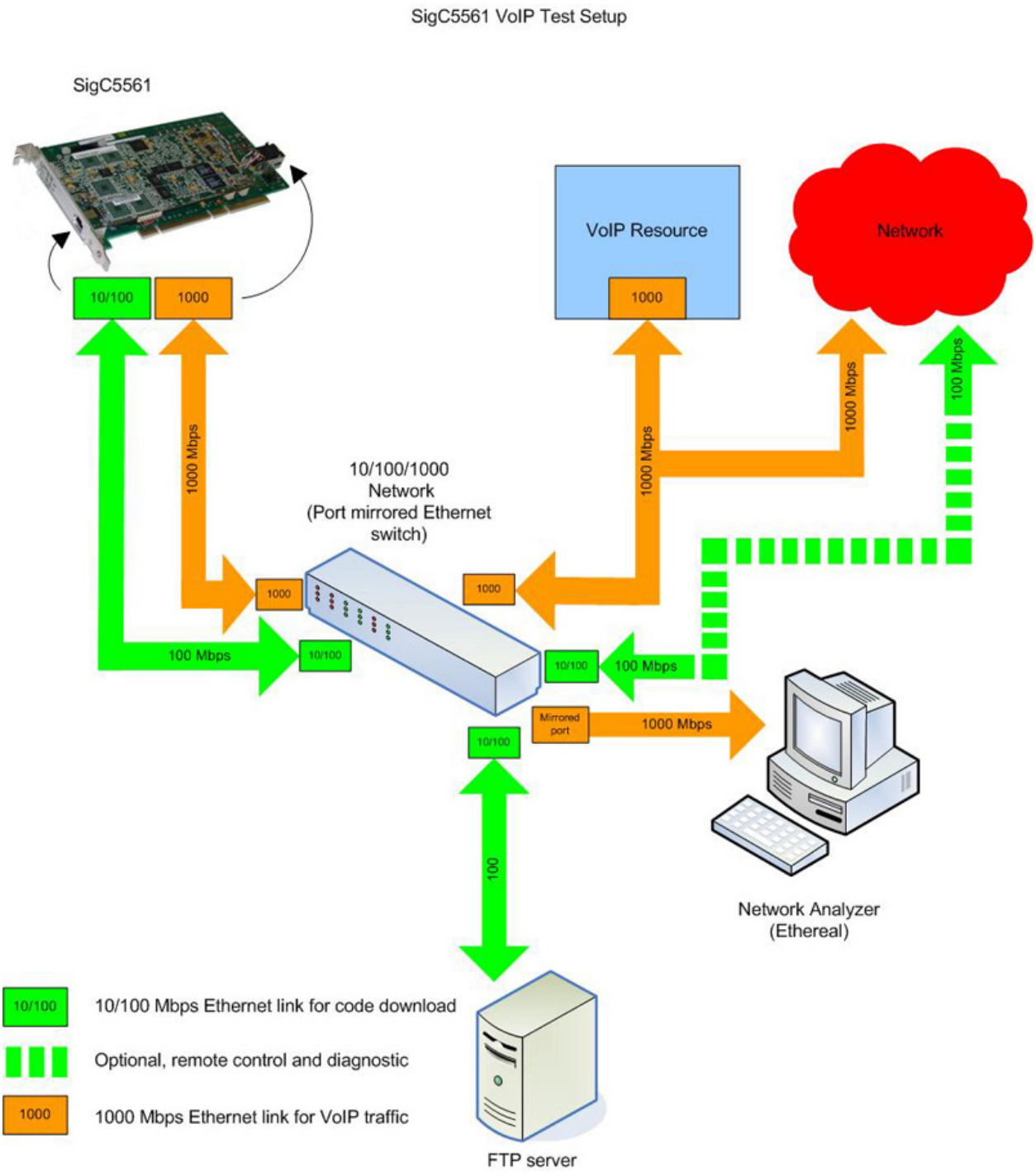

Figure 4-1 *SigC5561 VoIP Test Setup*

Below is a diagram that shows a second SigC5561 card used as the VoIP Resource:

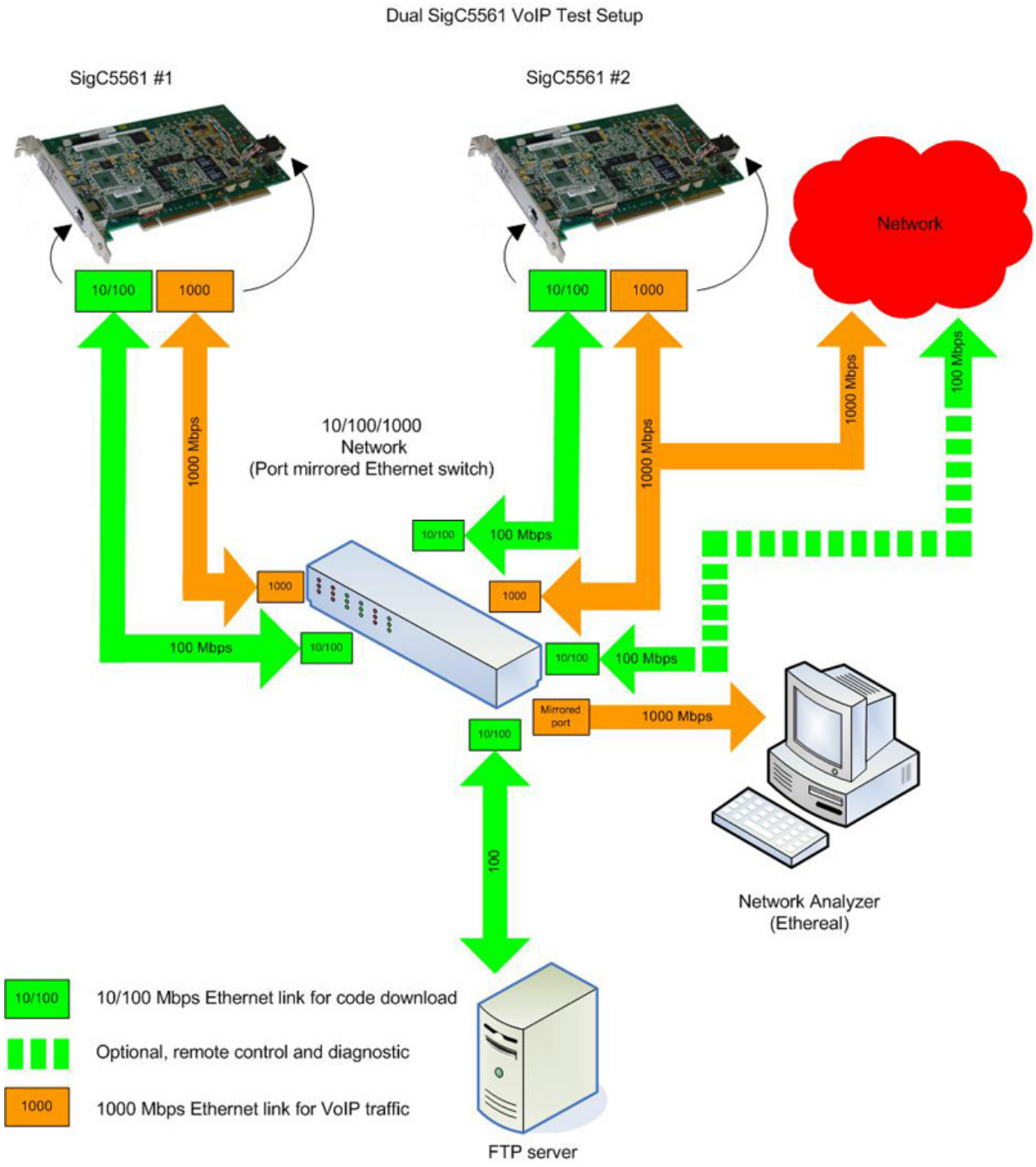

Figure 4-2 *Dual SigC5561 VoIP Test Setup*

The VoIP Resource does not need to have a high channel capacity, but to stress-test the SigC5561 product, at least 192 G711 channels (or 96 G729) channels are needed (assuming minimum 2-DSP configuration). The example in Figure 4-2 above is one way to perform stress-testing.

Below is a diagram that shows both a TDM Resource and VoIP Resource for a PCI card product:

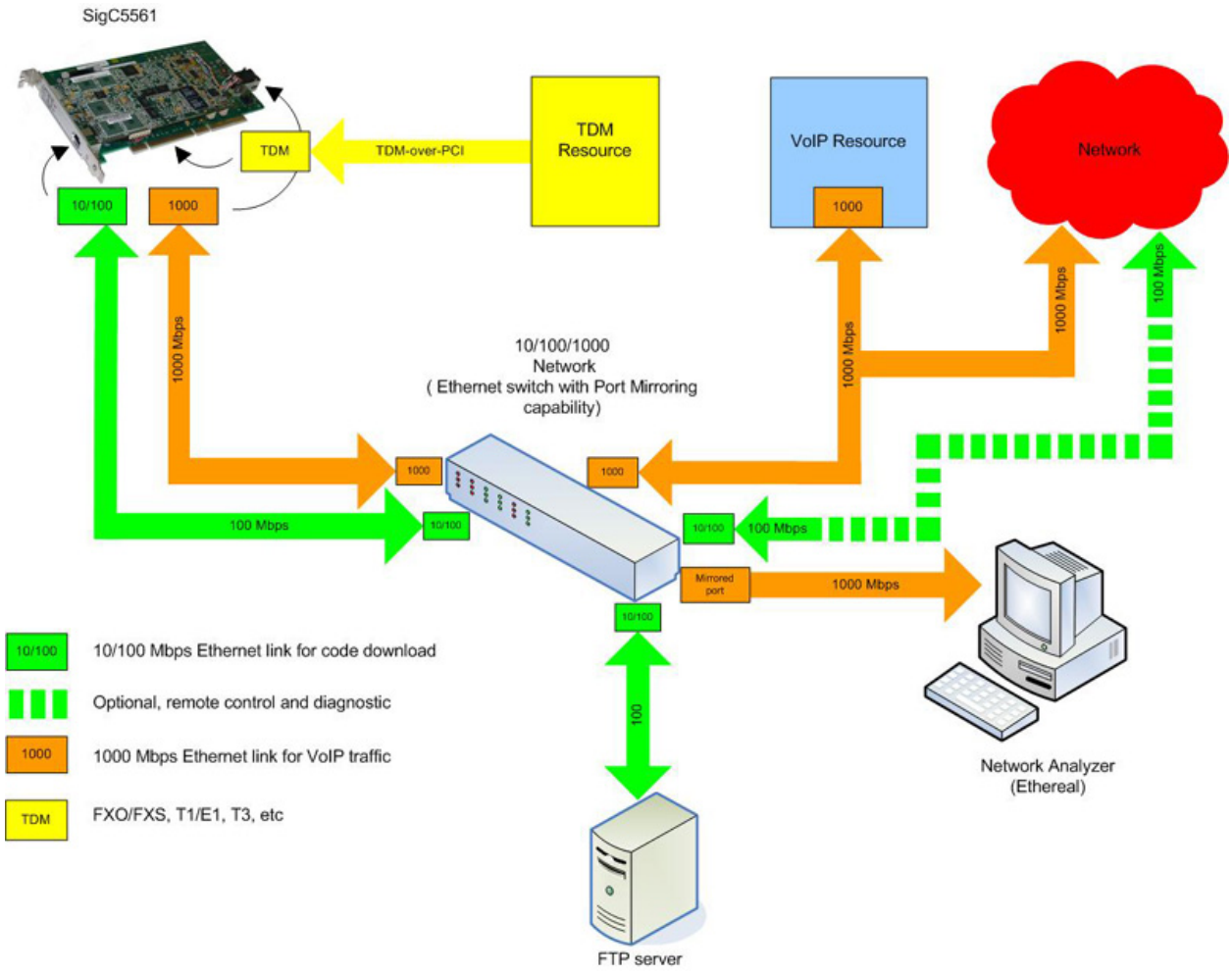

Figure 4-3 *SigC5561 PCI Card Test Setup Including TDM Resource*

Following are some comments about the above diagrams:

1. "VoIP Resource" can be just about any VoIP-capable equipment, including another SigC5561 product.
2. For initial test purposes, the SigC5561 and VoIP Resource should be together locally, for example on the same LAN in a lab environment.
3. The switch Mirror Port, although labeled as GbE, can also be 10/100.
4. If the 'VoIP Resource' is another SigC5561 card then the second card also needs its 10/100 port to be connected to a 10/100 port of the switch for FTP code download, as shown in Figure 4-2 above.
5. TDM-over-PCI data flow does not typically apply to SigC5561-PTMC modules, which support a dedicated TDM interface over a separate CTbus (H.110 subset) connector.

 Copyright © Signalogic 2006-2007

Before beginning test and checkout procedures, it's a good idea to double-check the following steps:

1. First, make sure hardware installation and cable connections have been made as described in section 2.0 above, Hardware Installation
2. Second, re-verify correct SigC5561 PCI card or PTMC module boot-up as described in section 2.3 above, Power-On

After making sure of the above steps, open a Linux command terminal to begin the test and checkout procedures listed below.

4.1 API Functional Test

Test programs, including source and makefiles, are included with SigC5561 software in order to test the SigC5561 API and also to provide "starting point" examples for application-specific programs.

Basic API test programs are called TestlibA2B, where A and B can be given as "TDM" or "IP". For example, to execute the TDM-to-IP test program, change the current folder to be:

```
/home/Signalogic/SigC5561/Test_Programs/testlib
```

and enter the command:

```
./TestlibTDM2IP –ips10.0.0.92 –ipd10.0.0.93 –n10 dnld –v -l
```

as shown in Figure 4-4 below. To avoid subsequent downloading of code to DSPs, exclude the 'dnld' identifier in the command above.

The '–ips' command-line option specifies IP *source address* for voice channels, and the '–ipd' command-line option specifies IP *destination address* for voice channels. In the test examples in this section, IP source and destination addresses are shown as 10.0.0.92 and 10.0.0.93 (local network addresses), but both values can be arbitrary. If the source and destination IP addresses are the same, then a loopback "self-test" is specified (note however, loopback is not recommended as an effective method to test the SigC5561 product).

The '-nN' command-line option opens N voice channels; where N is the number of channels. In the example show below, N is specified as 10 channels. A default value for N of one (1) is used if no –n option is specified.

The 'init' command-line option specifies that any previous SigC5561 output data still contained in buffer memory on the SigC5561 card (or module) will be displayed when the TestlibA2B program first runs. Normally this option would be omitted, but it can be useful. For example, if the SigC5561 card was recently booted, this option allows previous boot log status to be displayed.

The '-v' command-line option (verbose mode) enables application-level status message print-out.

The '-l' command-line option (loop mode) causes DSP code download status display to be updated per DSP.

root@localhost:/home/Signalogic/SigC5561/Test_Programs/testlib
[root@localhost testlib]# ./TestlibTDM2IP -ips10.0.0.92 -ipd10.0.0.93 -n10 dnld -v -l

Figure 4-4 *Running API Test Program "TestlibTDM2IP" with "dnld" Command-Line Option*

After running the TestlibA2B program, you should see a display similar to Figure 4-6 below. In this example, the display output indicates successful download of DSP code to the first DSP (DSP0) and the second DSP (DSP1) and successful open of 10 voice channels (N = 10).

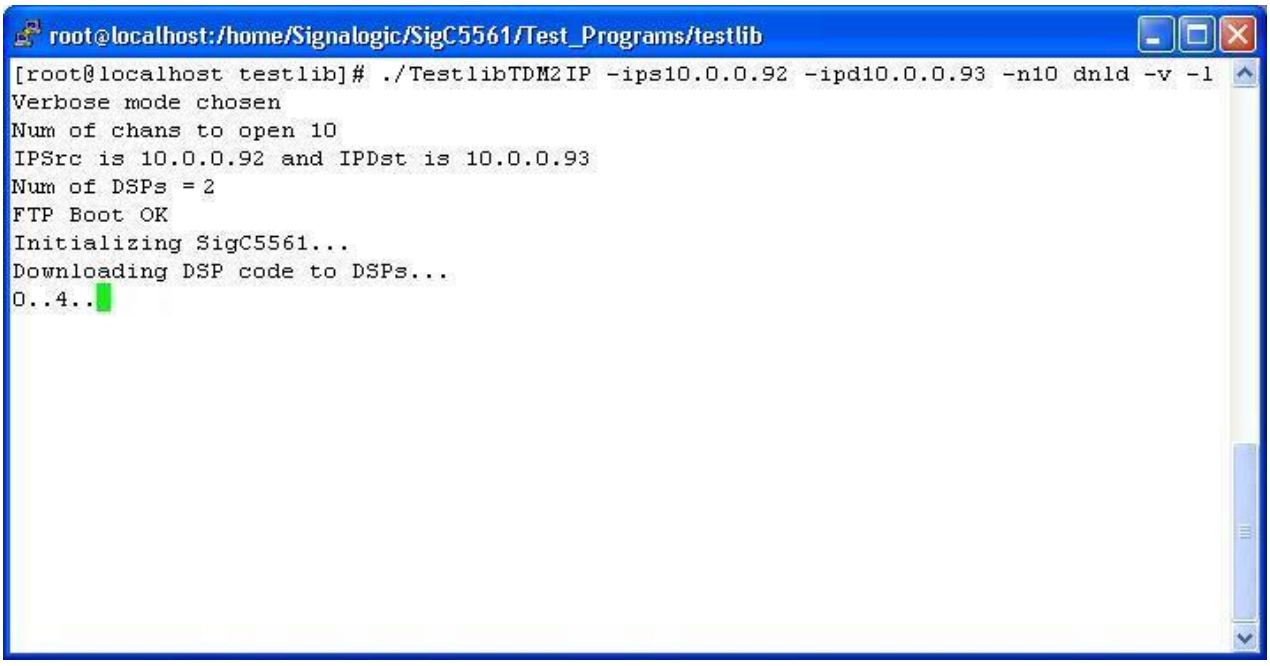
root@localhost:/home/Signalogic/SigC5561/Test_Programs/testlib
[root@localhost testlib]# ./TestlibTDM2IP -ips10.0.0.92 -ipd10.0.0.93 -n10 dnld -v -l
Verbose mode chosen
Num of chans to open 10
IPSrc is 10.0.0.92 and IPDst is 10.0.0.93
Num of DSPs = 2
FTP Boot OK
Initializing SigC5561...
Downloading DSP code to DSPs...
0..4..

Figure 4-5 *First Two DSPs Downloaded by API Test Program "TestlibTDM2IP"*

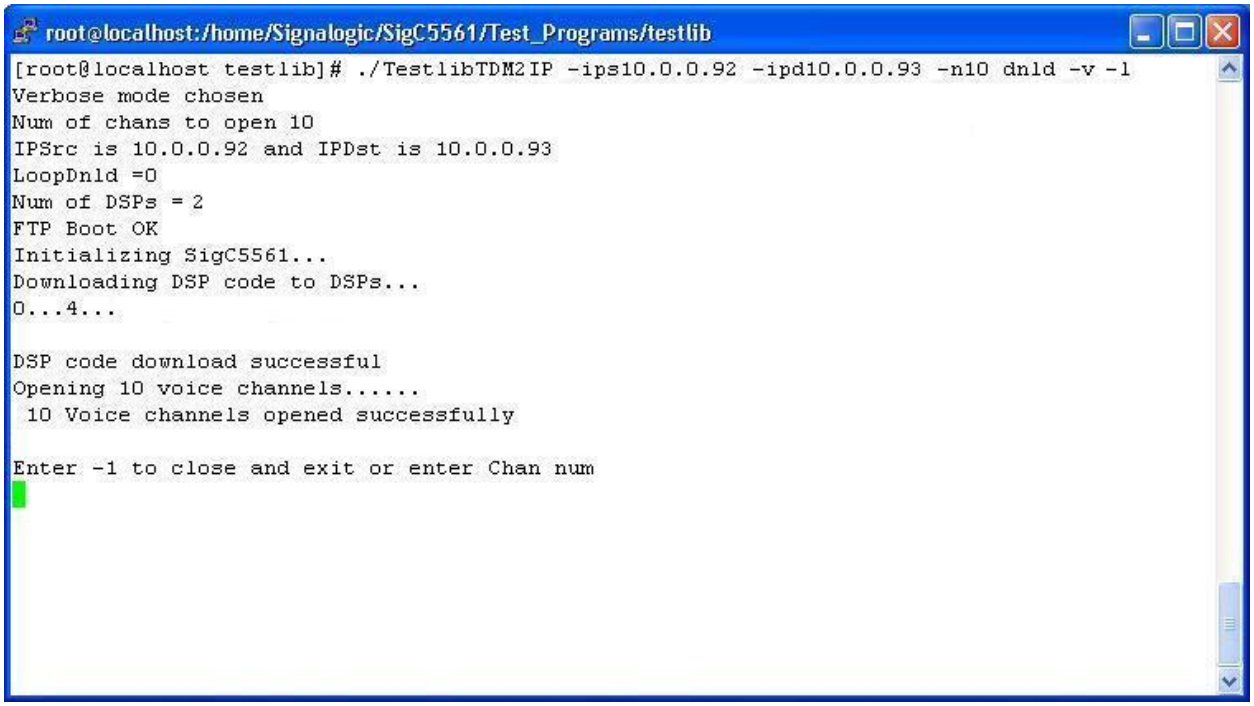
root@localhost:/home/Signalogic/SigC5561/Test_Programs/testlib
[root@localhost testlib]# ./TestlibTDM2IP -ips10.0.0.92 -ipd10.0.0.93 -n10 dnld -v -l
Verbose mode chosen
Num of chans to open 10
IPSrc is 10.0.0.92 and IPDst is 10.0.0.93
LoopDnld =0
Num of DSPs = 2
FTP Boot OK
Initializing SigC5561...
Downloading DSP code to DSPs...
0...4...
DSP code download successful
Opening 10 voice channels.....
10 Voice channels opened successfully
Enter -1 to close and exit or enter Chan num

Figure 4-6 *Successful Run of "TestlibTDM2IP" Program*

Note: Entering '-1' (minus one) will close the TestlibA2B programs. Entering a number from 0 to N-1 (where N is number of opened voice channels) will display statistics for the specified channel.

Figure 4-7 below gives an example for a 7-DSP download, with 2 channels opened, and no 'loop mode' command-line option:

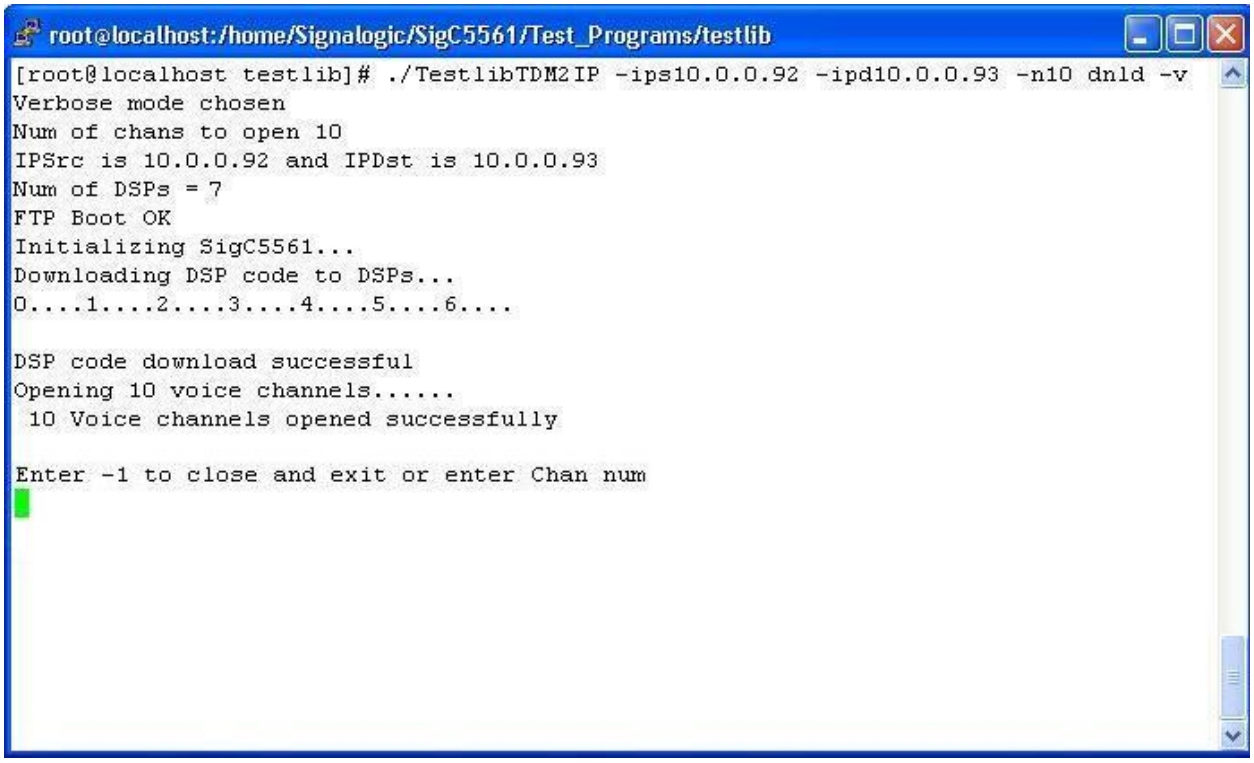
root@localhost:/home/Signalogic/SigC5561/Test_Programs/testlib
[root@localhost testlib]# ./TestlibTDM2IP -ips10.0.0.92 -ipd10.0.0.93 -n10 dnld -v
Verbose mode chosen
Num of chans to open 10
IPSrc is 10.0.0.92 and IPDst is 10.0.0.93
Num of DSPs = 7
FTP Boot OK
Initializing SigC5561...
Downloading DSP code to DSPs...
0....1....2....3....4....5....6....
DSP code download successful
Opening 10 voice channels......
10 Voice channels opened successfully
Enter -1 to close and exit or enter Chan num

Figure 4-7 *Successful Run of "TestlibTDM2IP" Program with 7-DSP Card*

Note in the examples above that a 2-DSP card does not load DSPs 0 and 1, but instead DSPs 0 and 4. This is due to the way DSPs populated for the 2-processor configuration at manufacturing time.

With the Testlib series program continuing to run, follow the steps below to proceed with SigC5561 card test and checkout:

1. Enter channel num 0 to N-1 to view statistics of currently open channels (where N is as defined in command-line options above), or enter '-1' to exit. Statistics for channel 0 are shown in Figure 4-8 below when '0' is entered.

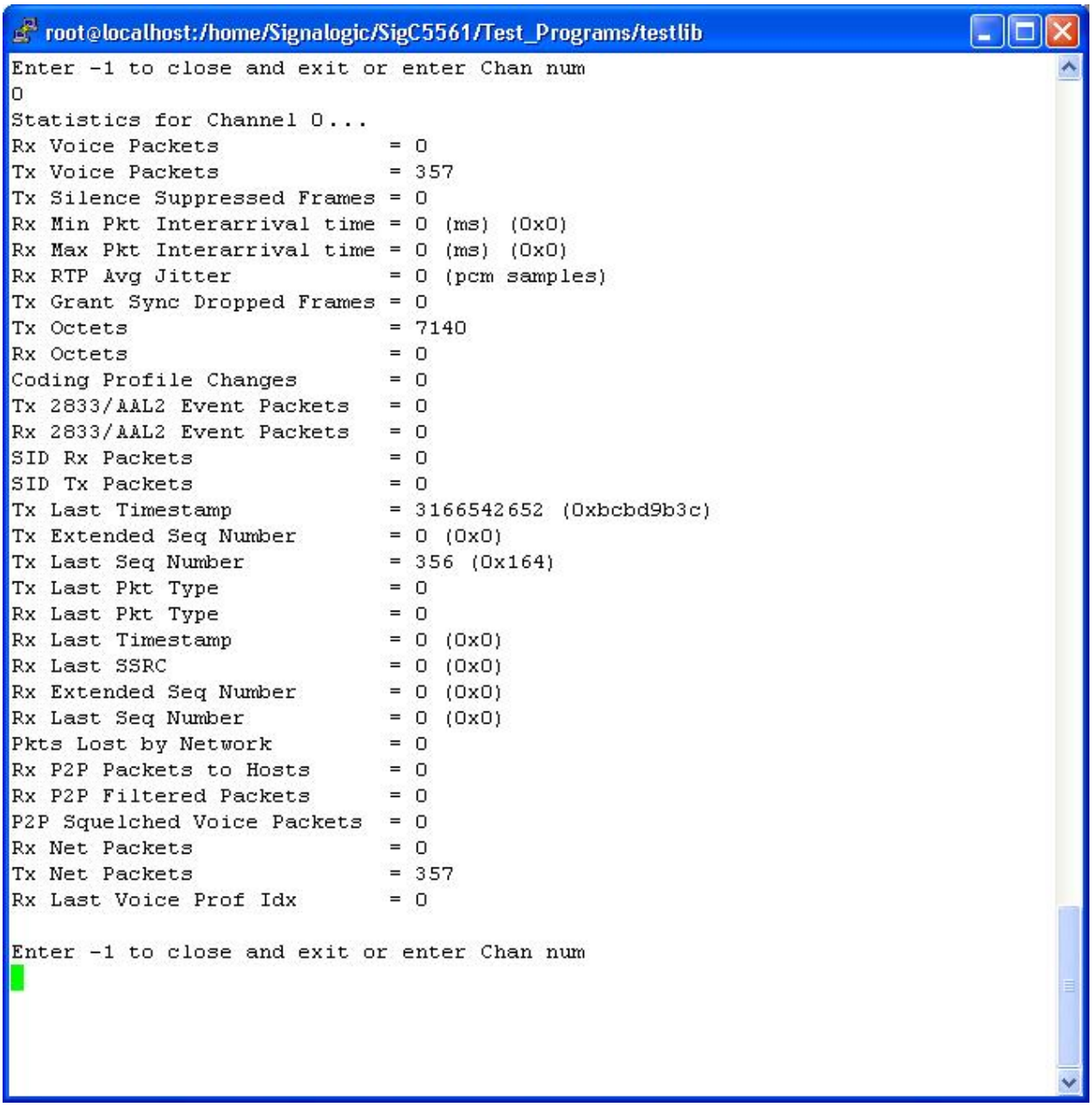
root@localhost:/home/Signalogic/SigC5561/Test_Programs/testlib
Enter -1 to close and exit or enter Chan num
0
Statistics for Channel 0...
Rx Voice Packets = 0
Tx Voice Packets = 357
Tx Silence Suppressed Frames = 0
Rx Min Pkt Interarrival time = 0 (ms) (0x0)
Rx Max Pkt Interarrival time = 0 (ms) (0x0)
Rx RTP Avg Jitter = 0 (pcm samples)
Tx Grant Sync Dropped Frames = 0
Tx Octets = 7140
Rx Octets = 0
Coding Profile Changes = 0
Tx 2833/AAL2 Event Packets = 0
Rx 2833/AAL2 Event Packets = 0
SID Rx Packets = 0
SID Tx Packets = 0
Tx Last Timestamp = 3166542652 (0xbcbd9b3c)
Tx Extended Seq Number = 0 (0x0)
Tx Last Seq Number = 356 (0x164)
Tx Last Pkt Type = 0
Rx Last Pkt Type = 0
Rx Last Timestamp = 0 (0x0)
Rx Last SSRC = 0 (0x0)
Rx Extended Seq Number = 0 (0x0)
Rx Last Seq Number = 0 (0x0)
Pkts Lost by Network = 0
Rx P2P Packets to Hosts = 0
Rx P2P Filtered Packets = 0
P2P Squelched Voice Packets = 0
Rx Net Packets = 0
Tx Net Packets = 357
Rx Last Voice Prof Idx = 0
Enter -1 to close and exit or enter Chan num

Figure 4-8 *Statistics Display for Channel 0*

2. Verify the GbE port of the SigC5561 is connected to a GbE switch with port mirroring capability, and the Mirror Port of the GbE switch is connected to a machine running the 'Ethereal' utility. Note that the PC running Ethereal should support GbE capability for best results (PCs older than 2003/2004 may not).

3. With the Testlib series program continuing to run; i.e. without entering '-1', please continue to section 4.3, Ethereal Packet Verification. If test results are satisfactory as indicated in section 4.3, then proceed to next step.

 Copyright © Signalogic 2006-2007

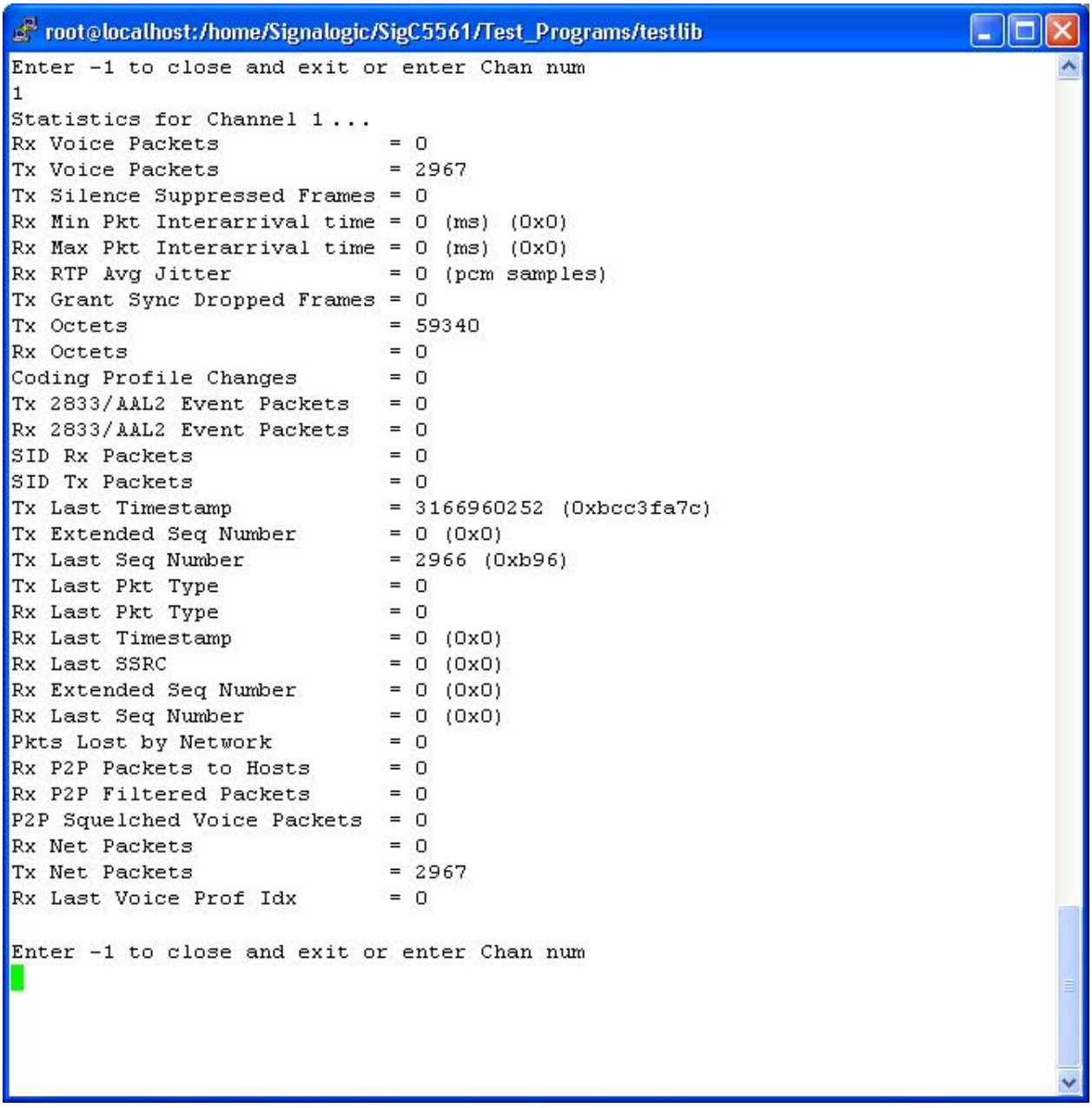
root@localhost:/home/Signalogic/SigC5561/Test_Programs/testlib
Enter -1 to close and exit or enter Chan num
1
Statistics for Channel 1 ...
Rx Voice Packets = 0
Tx Voice Packets = 2967
Tx Silence Suppressed Frames = 0
Rx Min Pkt Interarrival time = 0 (ms) (0x0)
Rx Max Pkt Interarrival time = 0 (ms) (0x0)
Rx RTP Avg Jitter = 0 (pcm samples)
Tx Grant Sync Dropped Frames = 0
Tx Octets = 59340
Rx Octets = 0
Coding Profile Changes = 0
Tx 2833/AAL2 Event Packets = 0
Rx 2833/AAL2 Event Packets = 0
SID Rx Packets = 0
SID Tx Packets = 0
Tx Last Timestamp = 3166960252 (0xbcc3fa7c)
Tx Extended Seq Number = 0 (0x0)
Tx Last Seq Number = 2966 (0xb96)
Tx Last Pkt Type = 0
Rx Last Pkt Type = 0
Rx Last Timestamp = 0 (0x0)
Rx Last SSRC = 0 (0x0)
Rx Extended Seq Number = 0 (0x0)
Rx Last Seq Number = 0 (0x0)
Pkts Lost by Network = 0
Rx P2P Packets to Hosts = 0
Rx P2P Filtered Packets = 0
P2P Squelched Voice Packets = 0
Rx Net Packets = 0
Tx Net Packets = 2967
Rx Last Voice Prof Idx = 0
Enter -1 to close and exit or enter Chan num

Figure 4-9 *Statistics Display for Channel 1*

4.2 PCI Terminal Program

NOTE: The following section assumes an FTP server is configured and running on the same network as connected to the SigC5561 product's 10/100 port, as explained in section 3.2.2 above, FTP Server Setup. Script files used in the examples below are expected to be located on the folder

```
C:\test
```

on the FTP server machine.

Go to the directory /home/Signalogic/SigC5561/Drivers. Load the driver module using the command 'make load'. If there is an error or the driver module is not already built (sigc55x.ko not present), please refer to the section on how to build the driver module.

1. Go to the test application folder /home/Signalogic/SigC5561/Test_Programs/terminal and execute the PCI terminal application program by typing './terminal init' as shown. The 'init' displays the entire contents of the buffer on start-up.

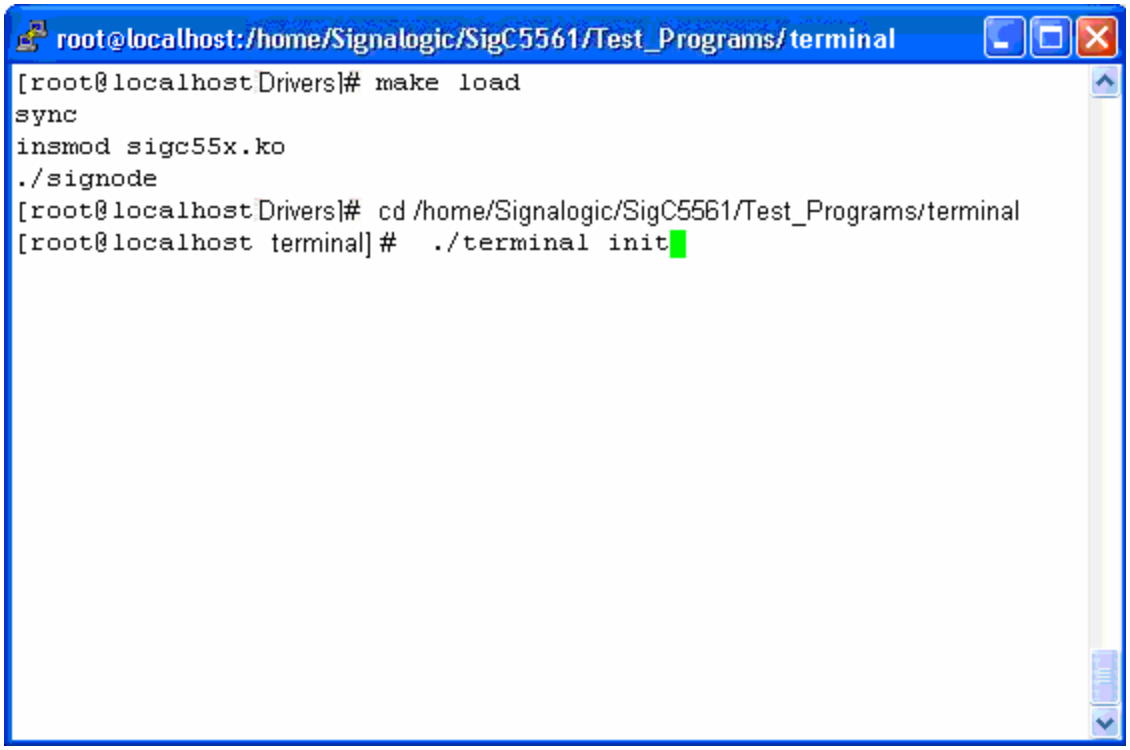

Figure 4-10 *'./terminal init' Command to Start PCI Terminal Test*

2. The start-up screen should look as below. Press 'ENTER' to get the '->'prompt

 Copyright © Signalogic 2006-2007

 Copyright © Signalogic 2006-2007

 Copyright © Signalogic 2006-2007

 Copyright © Signalogic 2006-2007

 Copyright © Signalogic 2006-2007

 Copyright © Signalogic 2006-2007

 Copyright © Signalogic 2006-2007

4.3 Ethereal Packet Verification

Ethereal analyzes the network information being captured over a network line. Ethereal is capable of displaying critical information about voice packets including the source and destination, time stamp, time duration interval of the packets being sent, etc…, over the designated monitoring network line. This information can be used for verification during the initial boot-up and anytime while the SigC5561 card is in use.

1. Connect the SigC5561 GbE port to a GbE switch port using CAT6 cable, and ensure the connection is reliable.
2. Connect the mirrored port of the switch to a separate PC running 'Ethereal' utility. ***This PC must have GbE capability.*** PCs older than 2003/2004 may not.
3. Start 'Ethereal'. Click on 'Capture' and select 'Interfaces' from the main menu bar.
4. You should see the packet count changing in the window, as shown in Figure 4-24 below.

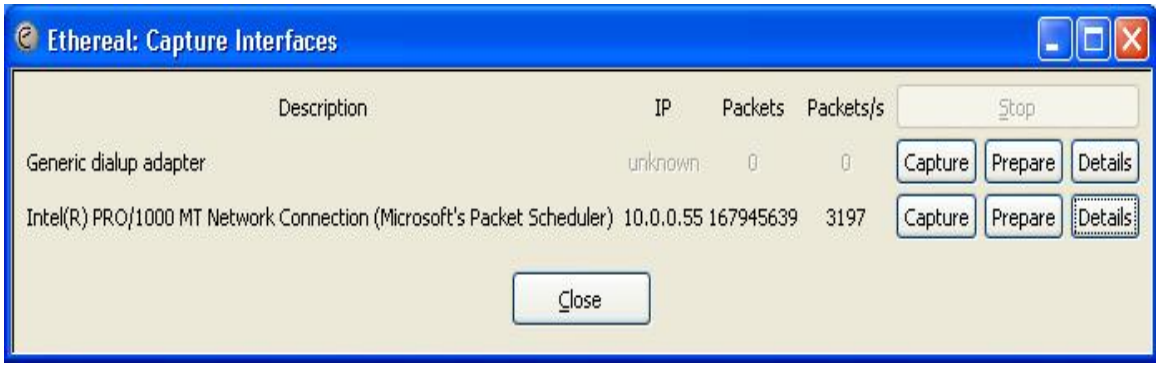

Figure 4-24 *Packet Count Window in Ethereal*

5. Click on 'Capture' button for the appropriate Ethernet device. The captured UDP packets would be displayed as below when you stop the capture. To avoid display of other packets filters may be set or disconnect the port mirroring switch from the external network. For specific packet filtering, such as to only monitor UDP traffic on port 32770, for example of the very first UDP port listed in the open channel script, enter 'udp.scrport == 32770' in the "**Filter:"** text box in picture shown below. Note that if Gigabit traffic is correct, the difference of packet time of No. 1 and No. 3 shown in picture below should be approximately 20 ms as it is in picture shown for codec G729.

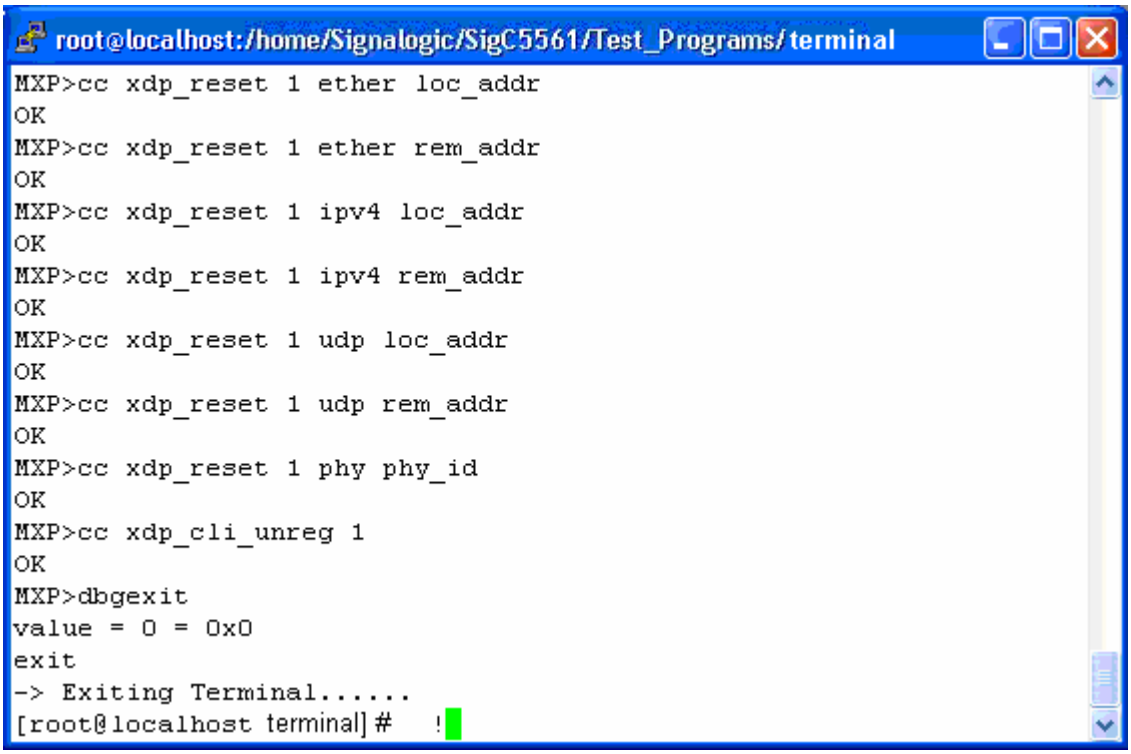

Figure 4-23 *Exiting the PCI Terminal Test*

 Copyright © Signalogic 2006-2007

4.3 Ethereal Packet Verification

Ethereal analyzes the network information being captured over a network line. Ethereal is capable of displaying critical information about voice packets including the source and destination, time stamp, time duration interval of the packets being sent, etc…, over the designated monitoring network line. This information can be used for verification during the initial boot-up and anytime while the SigC5561 card is in use.

1. Connect the SigC5561 GbE port to a GbE switch port using CAT6 cable, and ensure the connection is reliable.
2. Connect the mirrored port of the switch to a separate PC running 'Ethereal' utility. ***This PC must have GbE capability.*** PCs older than 2003/2004 may not.
3. Start 'Ethereal'. Click on 'Capture' and select 'Interfaces' from the main menu bar.
4. You should see the packet count changing in the window, as shown in Figure 4-24 below.

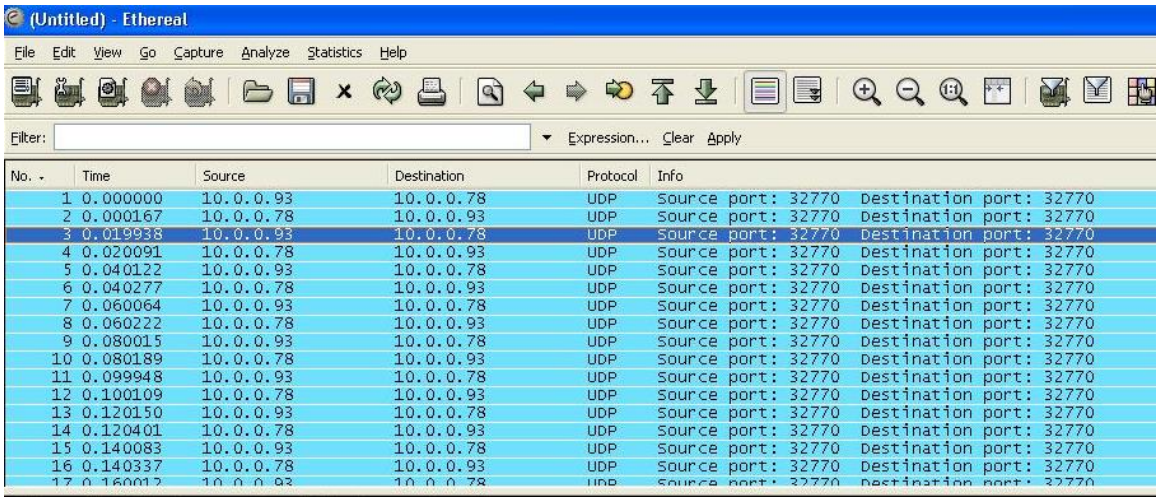
(Untitled) - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 10.0.0.93 10.0.0.78 UDP Source port: 32770 Destination port: 32770
2 0.000167 10.0.0.78 10.0.0.93 UDP Source port: 32770 Destination port: 32770
3 0.019938 10.0.0.93 10.0.0.78 UDP Source port: 32770 Destination port: 32770
4 0.020091 10.0.0.78 10.0.0.93 UDP Source port: 32770 Destination port: 32770
5 0.040122 10.0.0.93 10.0.0.78 UDP Source port: 32770 Destination port: 32770
6 0.040277 10.0.0.78 10.0.0.93 UDP Source port: 32770 Destination port: 32770
7 0.060064 10.0.0.93 10.0.0.78 UDP Source port: 32770 Destination port: 32770
8 0.060222 10.0.0.78 10.0.0.93 UDP Source port: 32770 Destination port: 32770
9 0.080015 10.0.0.93 10.0.0.78 UDP Source port: 32770 Destination port: 32770
10 0.080189 10.0.0.78 10.0.0.93 UDP Source port: 32770 Destination port: 32770
11 0.099948 10.0.0.93 10.0.0.78 UDP Source port: 32770 Destination port: 32770
12 0.100109 10.0.0.78 10.0.0.93 UDP Source port: 32770 Destination port: 32770
13 0.120150 10.0.0.93 10.0.0.78 UDP Source port: 32770 Destination port: 32770
14 0.120401 10.0.0.78 10.0.0.93 UDP Source port: 32770 Destination port: 32770
15 0.140083 10.0.0.93 10.0.0.78 UDP Source port: 32770 Destination port: 32770
16 0.140337 10.0.0.78 10.0.0.93 UDP Source port: 32770 Destination port: 32770

Figure 4-25 *UDP Packet Information Display in Ethereal*

6. Enter ‘0’ in the PCI terminal window to close the channels and exit the test program.

4.4 Troubleshooting

Use the following procedures to troubleshoot your system. If you have followed all of the procedures below and still need assistance, refer to the "Technical Support Procedures" section.

Note: Always disconnect the power cord before adding, changing or installing any cards, modules, and/or components.

No Power or LED Indicators

1. Make sure no short circuits exist in the systems and on the board set in and out of the system.
2. For the SigC5561 PCI card, verify the SigC5561 PMC connectors are completely inserted into the PMC mating connectors on the PCI-to-PMC Passive Adapter card. See section
3. Verify the board set (specifically the PCI-to-PMC Adapter) is completely inserted into a 3.3V PCI slot on the system's motherboard, and that the PCI slot on the mother board is a functioning 3.3V PCI slot.
4. Check that the 115V/230V switch on the power supply for the system is properly set and there is power cord connected from there to a power outlet.
5. Turn the power switch on the system on and off to test the system and board set.

No Power or LED indications on Gigabit Ethernet Switch

1. Make sure the GbE switch is power on.
2. Make sure the switch port connected to the SigC5561 is a working port.
3. Make sure the connection from switch port to SigC5561 GbE port is via CAT6 network cable.

Software Test Programs Fail

1. Verify FTP server activity at SigC5561 boot. Monitor the FTP log window and verify a connection is made, files are downloaded, etc.
2. Verify 33 MHz PCI bus clock. There may be a BIOS settings for "auto-detect" or 33 MHz override (either of those should work).

 Copyright © Signalogic 2006-2007

5 Platform Support and Configuration

This section covers support and configuration for different platforms.

5.1 Boot Configuration

5.1.1 FTP Download

*** Under Construction ***

5.1.2 Flash Load

*** Under Construction ***

5.2 Platform Support

The SigC5561-PTMC module supports the following platforms:

1. PCI slot (PCI specification 2.2)
2. cPCI carrier PMC or PTMC daughtercard site
3. ATCA carrier PMC or PTMC daughtercard site
4. Stand-Alone configuration

5.2.1 PCI Platform Support

Together, the PTMC module and PMC adapter make up the SigC5561 PCI card. Figure 5-1 below shows a SigC5561 card with 7-DSP configuration.

 Copyright © Signalogic 2006-2007

4.3 Ethereal Packet Verification

Ethereal analyzes the network information being captured over a network line. Ethereal is capable of displaying critical information about voice packets including the source and destination, time stamp, time duration interval of the packets being sent, etc…, over the designated monitoring network line. This information can be used for verification during the initial boot-up and anytime while the SigC5561 card is in use.

1. Connect the SigC5561 GbE port to a GbE switch port using CAT6 cable, and ensure the connection is reliable.
2. Connect the mirrored port of the switch to a separate PC running 'Ethereal' utility. ***This PC must have GbE capability.*** PCs older than 2003/2004 may not.
3. Start 'Ethereal'. Click on 'Capture' and select 'Interfaces' from the main menu bar.
4. You should see the packet count changing in the window, as shown in Figure 4-24 below.

_ilter:" text box in picture shown below. Note that if Gigabit traffic is correct, the difference of packet time of No. 1 and No. 3 shown in picture below should be approximately 20 ms as it is in picture shown for codec G729.

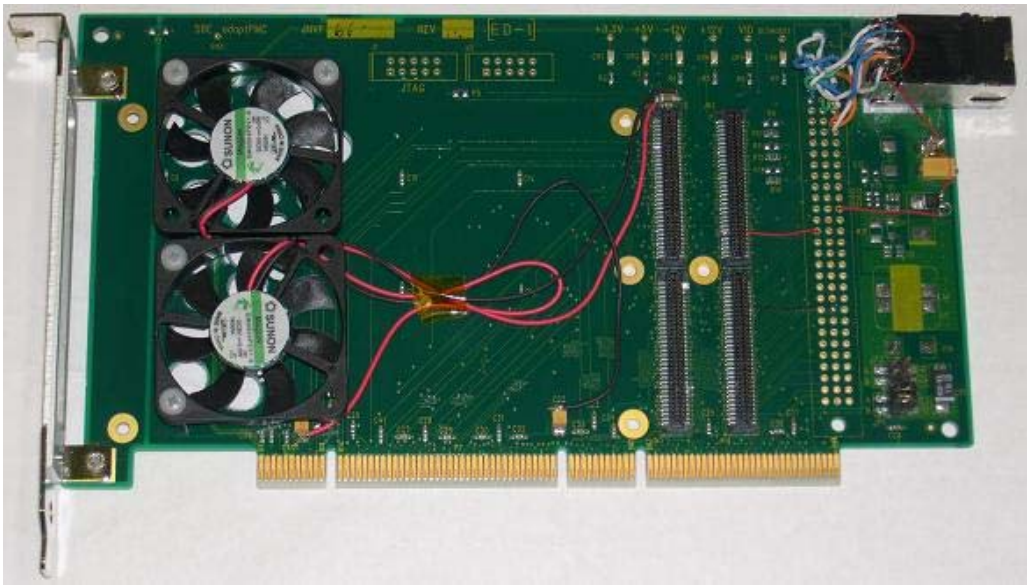

Figure 5-3 *PCI-to-PMC Passive Adapter*

For PCI card slot usage (i.e. PC and server platforms), the SigC5561-PTMC module and PCI-to-PMC adapter is delivered as a pre-assembled unit. For cPCI or ATCA carrier (blade) usage, the SigC5561-PTMC module is delivered separately, without an adapter card.

For a combined module + adapter card, it's strongly recommended to not disassemble the PTMC module and adapter card, unless properly instructed to do so by qualified Signalogic technical support personnel. As noted above, if disassembly is attempted without prior approval of qualified Signalogic technical support personnel, the product warranty will be voided.

For explanation purposes only, assembly of the PTMC module and adapter card is shown below.

If disassembling the SigC5561 card, the metal screws attaching the SigC5561-PTMC module to the adapter card must be removed first.

If assembling the SigC5561 card, reference Figure 5-4 below for proper gripping of the SigC5561-PTMC module. First insert the bezel end of the SigC5561-PTMC module into the passive adapter card bracket's opening at approximately a 35 to 45 degree angle, as shown in Figure 5-5and Figure 5-6 below (this step labeled as arrow #1). Then slowly and gently lower the PMC end of the SigC5561-PTMC module down to the PMC connectors on the passive adapter card (this step labeled as arrow #2 in Figure 5-5).

 Copyright © Signalogic 2006-2007

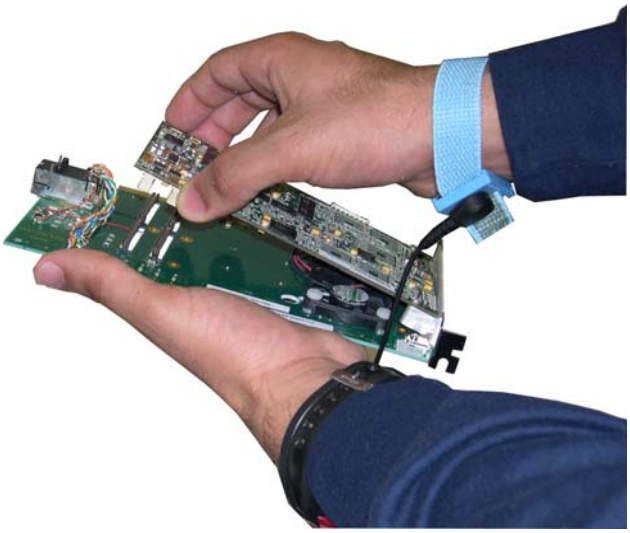

Figure 5-4 *Proper Grip for Handling PTMC Module During Assembly*

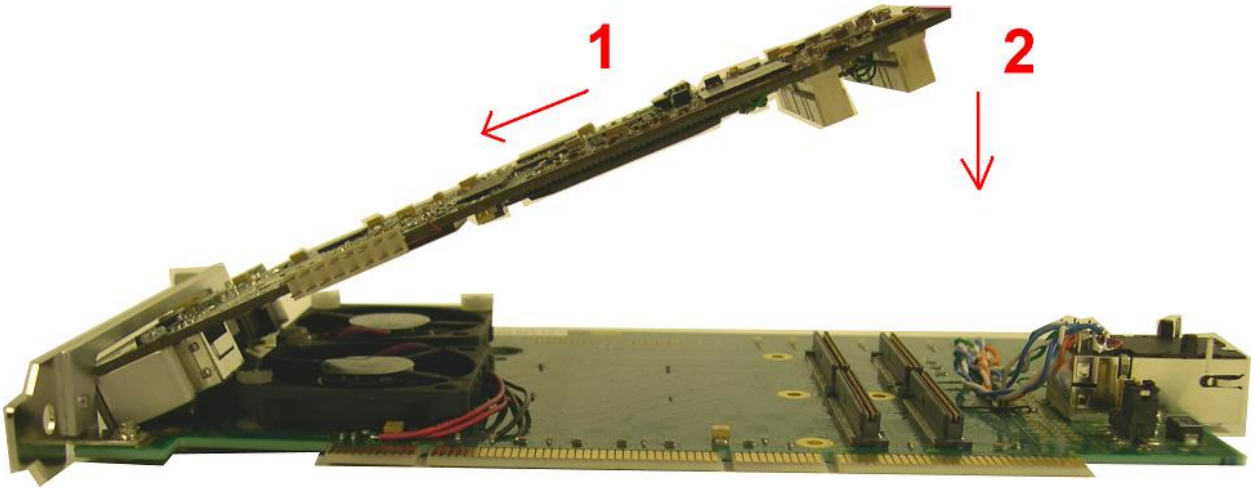

Figure 5-5 *Mating SigC5561-PTMC Module to Passive Adapter Card, Side View*

Figure 5-6 *Mating SigC5561-PTMC Module to Passive Adapter Card, Top View*

Then gently push down at top corners of the card to insert the PMC connectors until there is no noticeable gap between the PMC connectors, as shown in Figure 5-7 below.

Figure 5-7 ***CORRECT*** *-- SigC5561-PTMC Module PMC Connectors Fully Seated*

Figure 5-8 below shows the SigC5561-PTMC module incorrectly seated on the passive adapter card. Plugging in the card and applying power in this situation will damage the SigC5561-PTMC module.

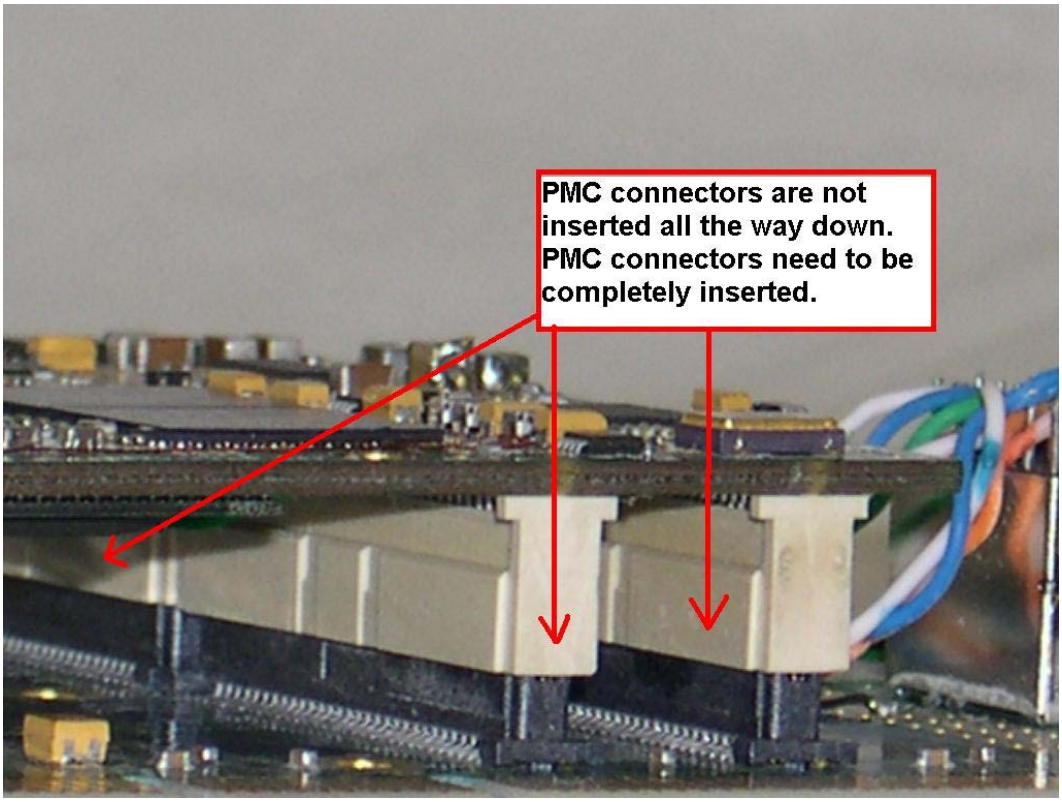

Figure 5-8 ***WRONG*** -- *SigC5561-PTMC Module PMC Connectors Show Noticeable Gap, Not Fully Seated*

5.2.2 cPCI Platform Support

*** This section under construction ***

For cPCI carrier (blade) usage, the SigC5561-PTMC module is delivered with or without a Debug Tab. For proper installation/removal of the SigC5561-PTMC module to and from the cPCI carrier card, please see section 2.2.6 above.

5.2.3 ATCA Platform Support

*** This section under construction ***

5.2.4 Debug Tab

For application-specific code development and debug, a SigC5561-PTMC module with attached Debug Tab may be used to allow access to low-level module data. A SigC5561-PTMC module with Debug Tab is shown installed on a cPCI carrier board in Figure 5-9 below.

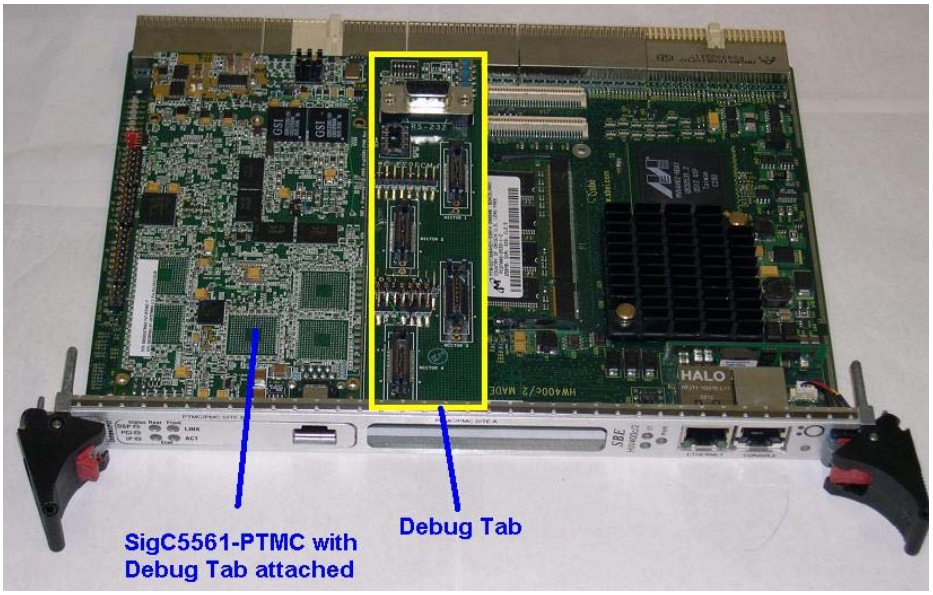

Figure 5-9 ***SigC5561-PTMC Module with Debug Tab Installed on cPCI Carrier Board***

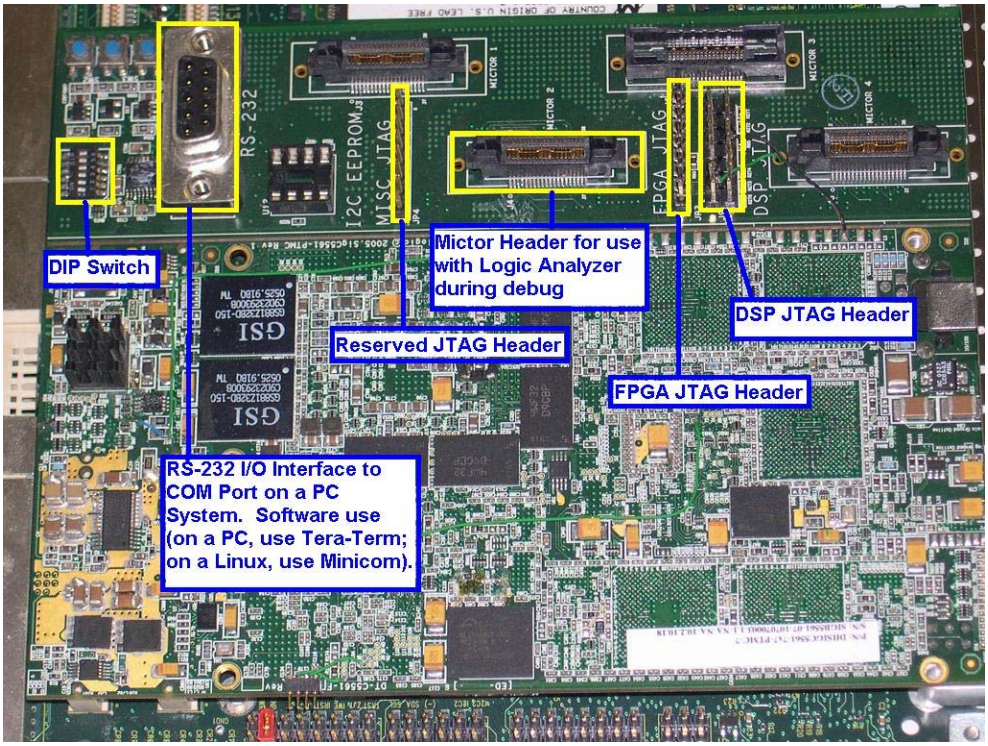
DIP Switch
Mictor Header for use with Logic Analyzer during debug
Reserved JTAG Header
DSP JTAG Header
FPGA JTAG Header
RS-232 I/O Interface to COM Port on a PC System. Software use (on a PC, use Tera-Term; on a Linux, use Minicom).
RS-232
I2C EEPROM
MISC JTAG
FPGA JTAG
DSP JTAG

Figure 5-10 ***Debug Tab Layout***

 Copyright © Signalogic 2006-2007

Debug Tab components are highlighted in Figure 5-10 above, and described in the table below.

Debug Tab Components	Description	Configuration Options
SW1	DIP Switch to set values on PTID [2:0] for supported PTMC Configurations Switch positions 1, 2, and 3 not used	DIP Switch Postions [4 5 6] (0 = off, 1 = on) [1 0 1] PTMC 2 [0 0 1] PTMC 3 [0 1 0] PTMC 5 Note if the Debug Tab is not present, then zero-ohm pull-up / pull-down resistors are available on the module to set the PTMC configuration
RS-232 DB9 Connector	Allows RS-232 terminal I/O (TTY) interface between host PC and VxWorks shell running on WinPath network processor	Recommended Host COM Port Settings Port = COM1 or COM2 Baud Rate = 115200 Data = 8 bit Parity = None Stop = 1 bit Flow Control = None For WinXP machines, Hyperterminal or TeraTerm are recommended. For Linux machines, Minicom is recommended If a slower baud rate is required, please ask Signalogic tech support for assistance
Mictor Headers J3, J4, J5, J6	Mictor Headers can be connected to a Logic Analyzer and used for low-level signal debug ***Note: The Logic Analyzer interface requires a Verilog source license agreement with Signalogic***	Defined in other documentation
Misc. JTAG Header	JTAG Header for WinPath network processor and other network devices ***Note: This interface requires a license agreement with Signalogic***	Pin Assignment 1 = VCC (3.3V) 2 = GND 3 = TCK 4 = TDO 5 = TDI 6 = TMS 7 = TRST 8 = RST
DSP JTAG Header	JTAG Header for DSP JTAG Chain ***Note: The DSP JTAG***	Pin Assignment 1 = TMS 2 = TRST

	interface requires separate license agreements with Texas Instruments and Signalogic	3 = TDI 4 = GND 5 = VCC (3.3V) 6 = NO CONNECT 7 = TDO 8 = GND 9 = TCK_RET 10 = GND 11 = TCK 12 = GND 13 = EMU0 14 = EMU1
FPGA JTAG Header	JTAG Header use for programming and debugging FPGAs ***Note: The FPGA JTAG interface requires a Verilog source license agreement with Signalogic***	Pin Assignment 1 = VCC (3.3V) 2 = GND 3 = TCK 4 = N/C 5 = TDO 6 = TDI 7 = TMS 8 = PROG_EN

Note that for a PCI card configuration, if the Debug Tab is used then a PCI extender is required to allow space between the tab and motherboard. No extenders of any type are required for cPCI and ATCA carrier boards, although it may be necessary to remove one or more other PTMC modules to avoid mechanical conflict with the SigC5561 Debug Tab.

5.2.5 Stand-Alone Configuration

*** This section under construction ***

 Copyright © Signalogic 2006-2007

6 Building Host Application Software

In this section are some examples of building applications that use DirectDSP software and the DSPower-MGTS voice and TDM extensions to DirectDSP.

DirectDSP software is an intermediate library that interfaces host application level software to DSP hardware. On its "API side", DirectDSP software presents a well-documented, consistent series of function calls to C/C++ programs under Linux, and C/C++, Visual Basic, and MATLAB programs under WinXP. On its "driver side", DirectDSP software interfaces with Linux and WinXP drivers that allow access directly to voice, video, wireless, and general DSP hardware resources. At the hardware level, driver code typically accesses the HPI (host port interface) peripheral on the DSP device(s) and dual-port memory interface on the network processor in order to implement real-time control and data transfer over the PCI bus.

Following are some examples of functionality provided by combined DirectDSP and DSPower-MGTS software API calls:

- Initialize SigC5561 hardware and DSP devices
- Download DSP code builds (ITU G.7xx, GSM, EVRC, etc) to one or more DSPs as needed
- Open TDM-to-IP, IP-to-IP, data (fax), or signaling voice channels
- Open conferencing bridges and specify nodes (voice channels)
- Set channel parameters for echo cancellation, RTP, tone detection, VAD / CNG, DTMF, and fax relay
- Get channel statistics, get DSP and other resource statistics, get voice quality statistics

DirectDSP fully supports multiprocessor hardware, including multiprocessor DSP code download, device control, and onchip memory access.

6.1 Rebuilding SigC5561 LKM Driver

This section describes how to rebuild LKM (loadable kernel module) drivers used by SigC5561 PCI card and PTMC module under Linux.

Note that LKM drivers should always be rebuilt when installing hardware or re-building the kernel, to avoid mismatches that could affect compatibility between software components.

Following are instructions for rebuilding the SigC5561 driver:

1) Log in as root
2) To make the driver module for your system, an appropriate Makefile is needed.
    a. For Red Hat,

    If Red Hat with kernel version 2.4.xx is installed on your system, then change the working directory to be:

    ```
    /home/Signalogic/SigC5561/Drivers/Makefiles/RedHat_24
    ```

    Do a ‘ls’ command to see the available Makefiles under the RedHat_24 directory. Copy the file ‘Makefile_2.4.18’ to the Drivers folder as file name ‘Makefile’ with command ‘cp Makefile_2.4.18 ../../Makefile’ as shown below. The kernel version in the file name indicates the supported Red Hat kernel version.

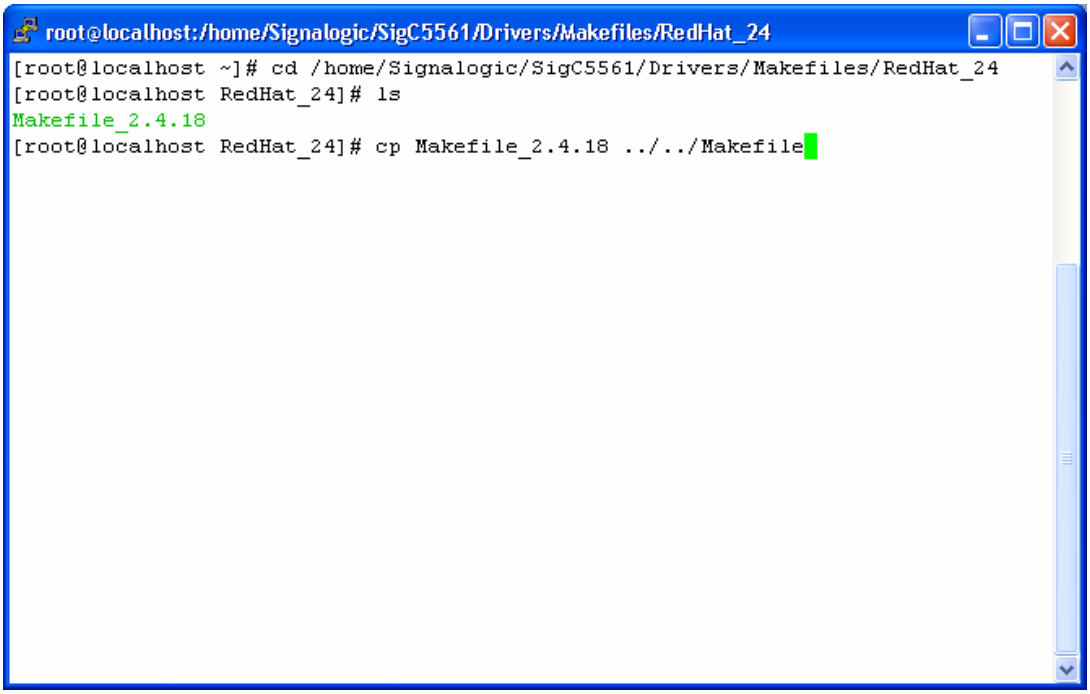

Figure 6-1 Get the RedHat 2.4 *SigC5561 Driver Module Makefile*

b. For CentOS,

If CentOS with kernel version 2.6.xx is installed on your system, then change the working directory to be:

```
/home/Signalogic/SigC5561/Drivers/Makefiles/CentOS_26
```

Do a ‘ls’ command to see the available Makefiles under the CentOS_26 directory. Copy the file ‘Makefile_2.6.9’ to the Drivers folder as file name ‘Makefile’ with command ‘cp Makefile_2.6.9 ../../Makefile’ as shown below. The kernel version in the file name indicates the supported CentOS kernel version for the Makefile.

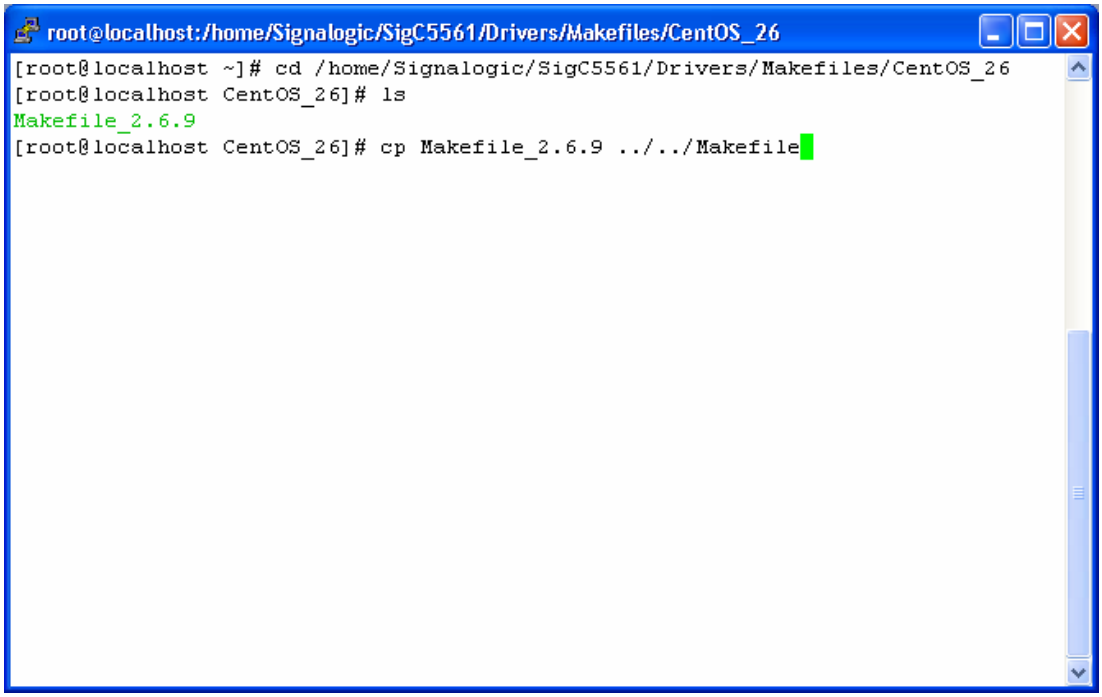

Figure 6-2 Get the CentOS 2.6 *SigC5561 Driver Module Makefile*

3) Change the working directory to be:

```
/home/Signalogic/SigC5561/Drivers
```

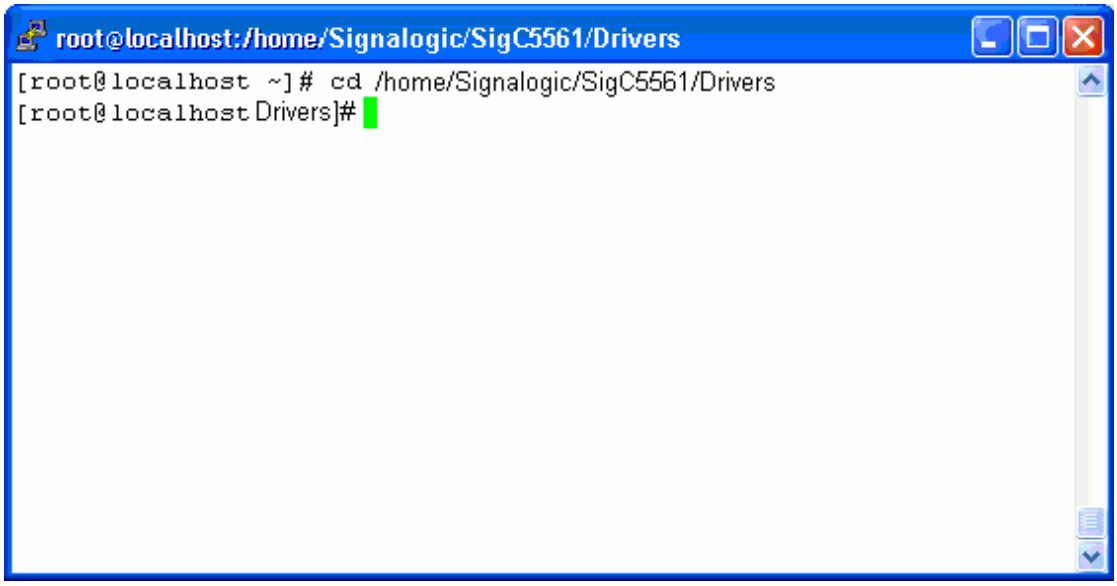

Figure 6-3 *SigC5561 Driver Module Source Location*

Build and load the driver by executing the following commands:

```
make clean
make
make load
```

After executing the above command, you should see a display similar to Figure 6-4 below:

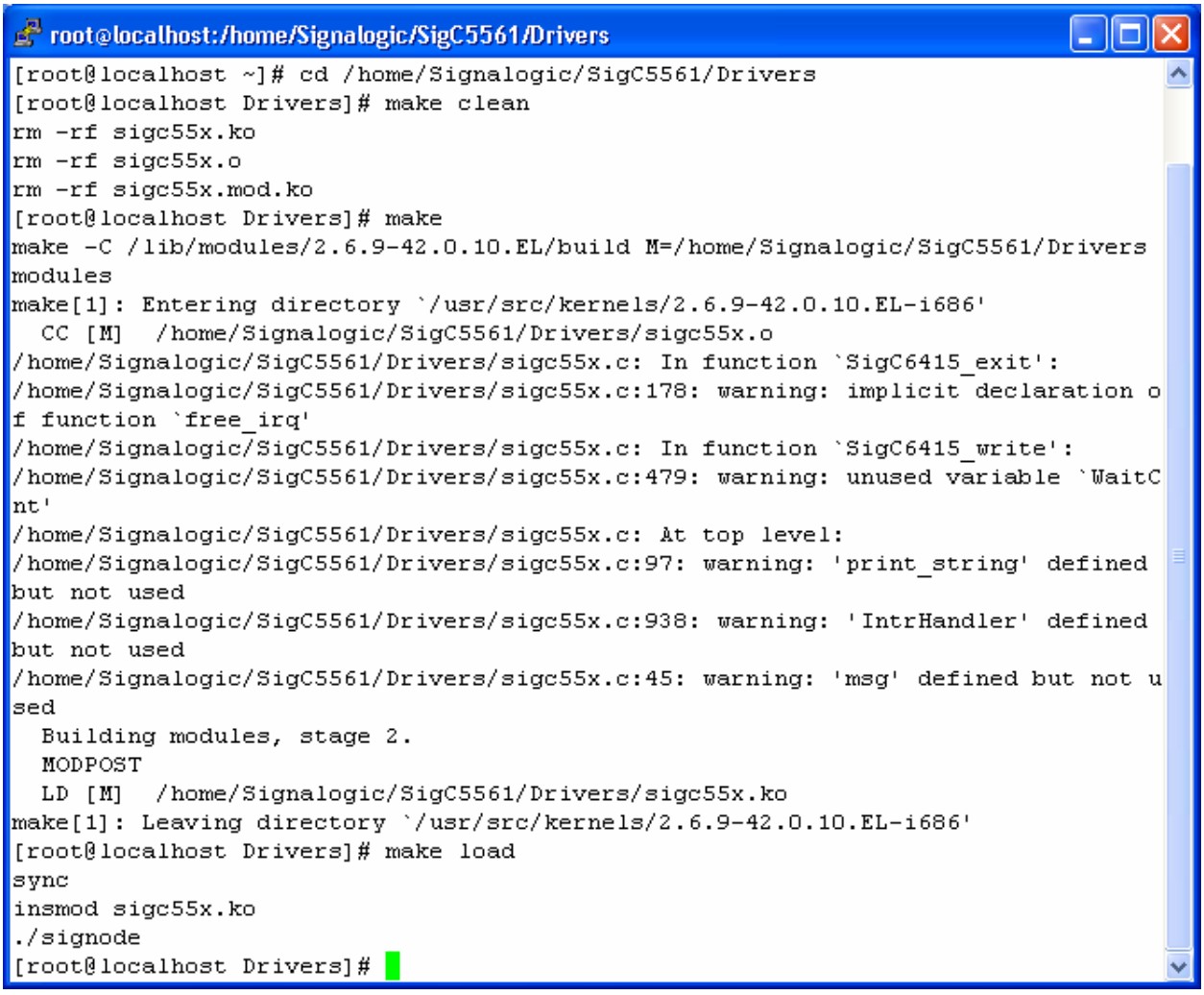

Figure 6-4 *Building and Loading SigC5561 Driver Module*

5) To verify if the driver module has been loaded correctly, enter the following command:

```
lsmod
```

You should see a module named ‘sigc55x’ in the list, as shown in Figure 6-5 below.

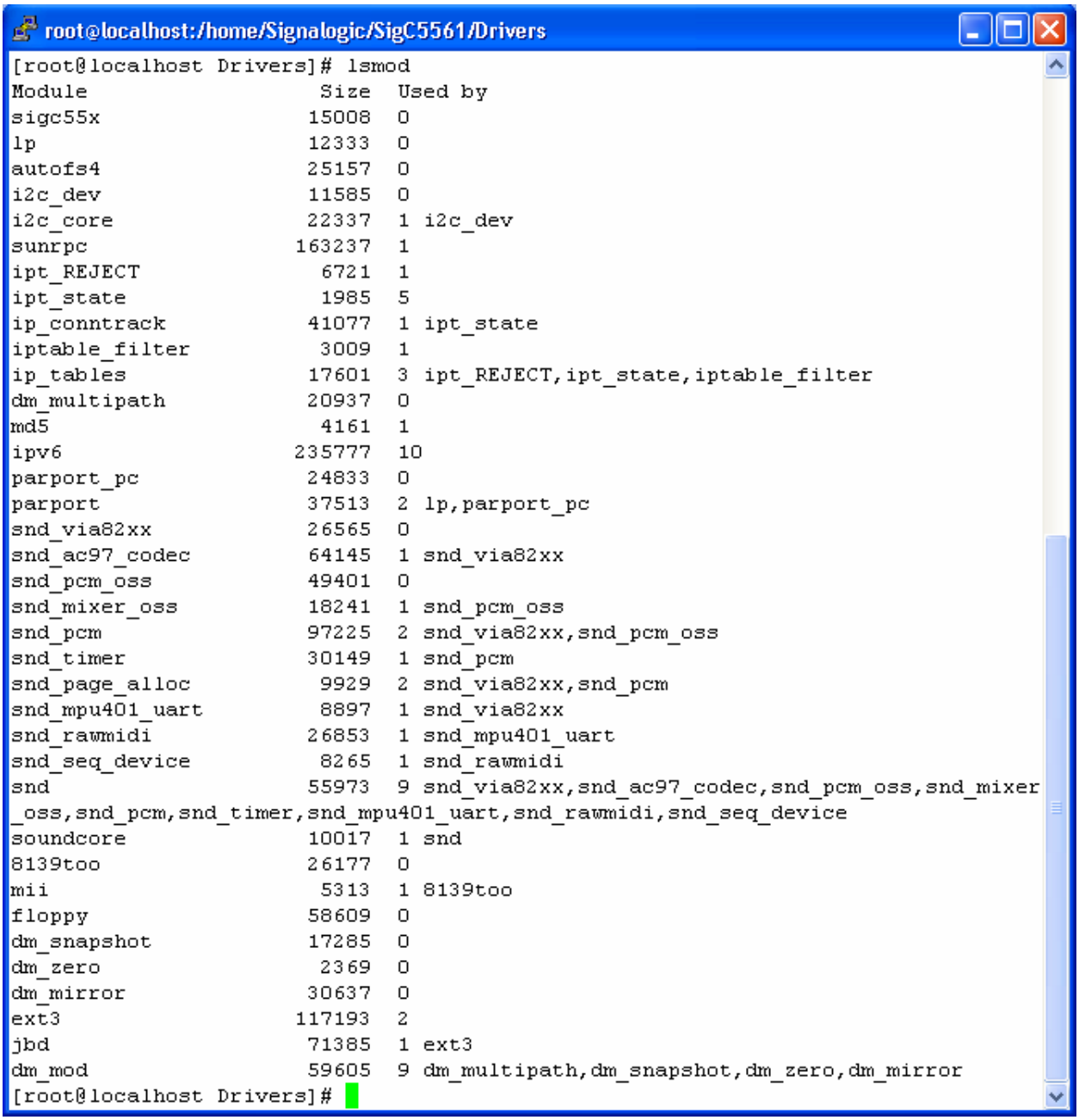

Figure 6-5 *Verifying SigC5561 Driver Source Build and Module Load*

6) To remove the module, enter the command:

```
rmmod sigc55x
```

Again, verify with the ‘lsmod’ command. The module may be reloaded using the command ‘make load’ again from /home/Signalogic/SigC5561/Drivers directory.

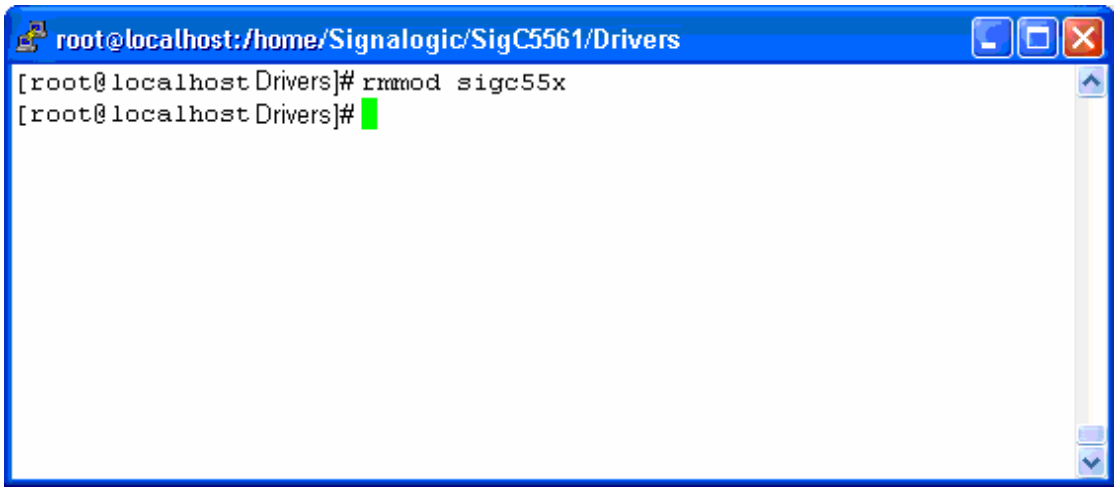

Figure 6-6 *Removing the SigC5561 Driver Module*

7) Note an error occurs if the sigc55x module is already loaded and an attempt is made to load it again using ‘make load’ without removing it first, as shown below.

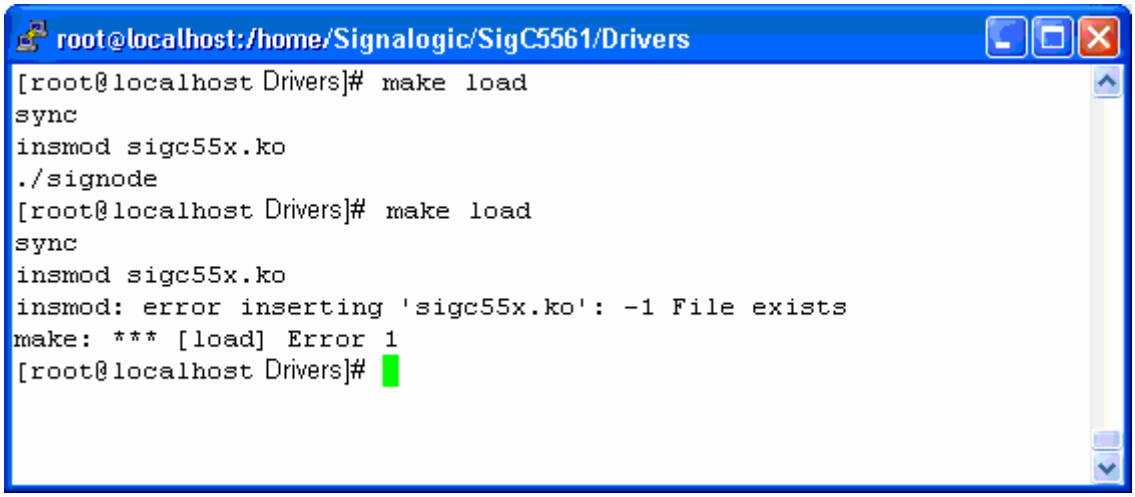

Figure 6-7 *Attempting to Double-Load the SigC5561 Driver*

6.2 Rebuilding Test Programs

Following are instructions for rebuilding the SigC5561 “TestlibA2B” series of programs (see also section 4.1 above, API Test Functionality).

1) Log in as root
2) Change working directory to be:

```
/home/Signalogic/SigC5561/Test_Programs/testlib
```

3) Build the TestlibA2B series program by executing the following commands:

```
make clean
make TestlibA2B
```

where A and B are “TDM” or “IP”, for example:

```
make clean
make TestlibTDM2IP
```

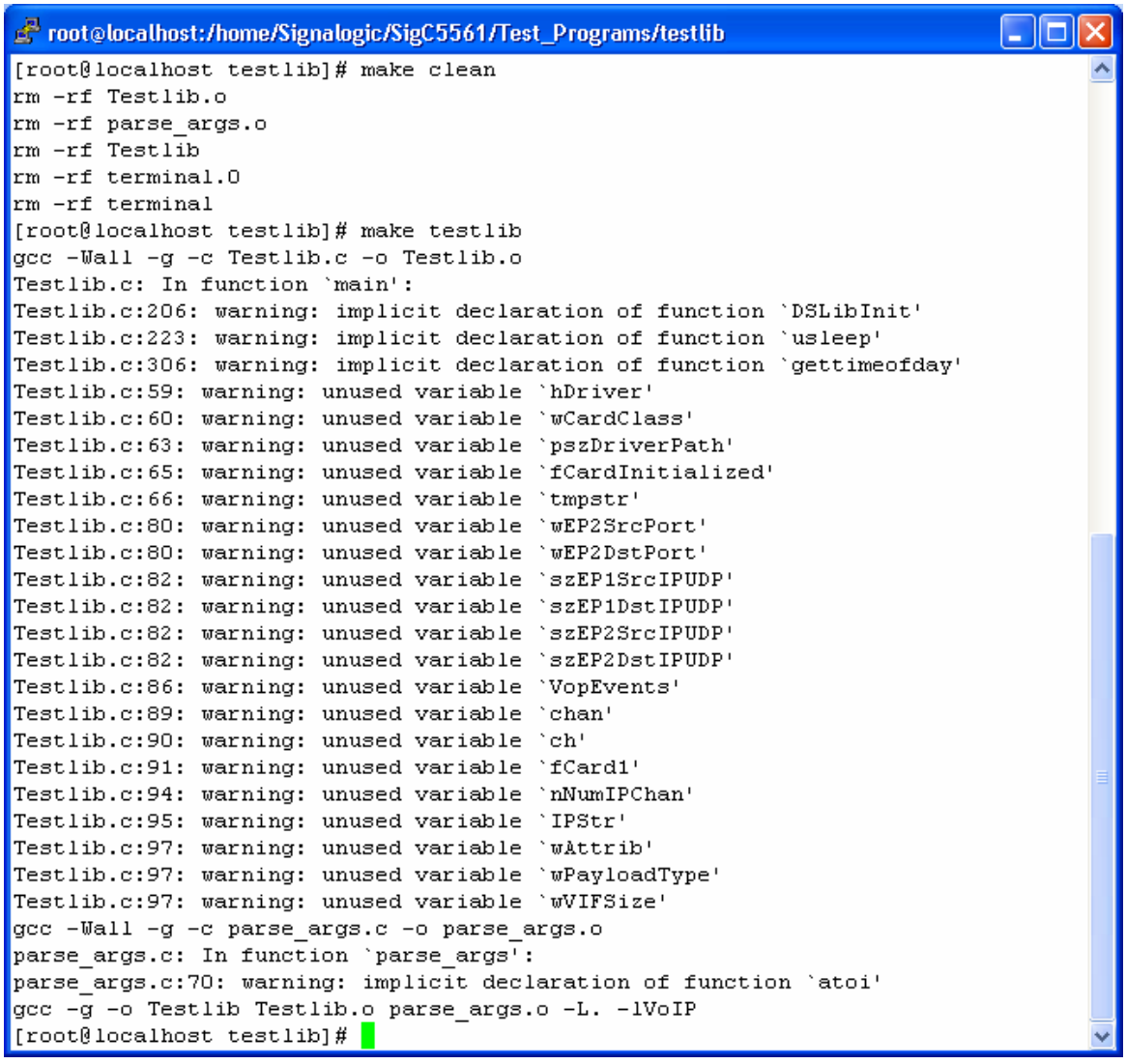
root@localhost:/home/Signalogic/SigC5561/Test_Programs/testlib
[root@localhost testlib]# make clean
rm -rf Testlib.o
rm -rf parse_args.o
rm -rf Testlib
rm -rf terminal.O
rm -rf terminal
[root@localhost testlib]# make testlib
gcc -Wall -g -c Testlib.c -o Testlib.o
Testlib.c: In function `main':
Testlib.c:206: warning: implicit declaration of function `DSLibInit'
Testlib.c:223: warning: implicit declaration of function `usleep'
Testlib.c:306: warning: implicit declaration of function `gettimeofday'
Testlib.c:59: warning: unused variable `hDriver'
Testlib.c:60: warning: unused variable `wCardClass'
Testlib.c:63: warning: unused variable `pszDriverPath'
Testlib.c:65: warning: unused variable `fCardInitialized'
Testlib.c:66: warning: unused variable `tmpstr'
Testlib.c:80: warning: unused variable `wEP2SrcPort'
Testlib.c:80: warning: unused variable `wEP2DstPort'
Testlib.c:82: warning: unused variable `szEP1SrcIPUDP'
Testlib.c:82: warning: unused variable `szEP1DstIPUDP'
Testlib.c:82: warning: unused variable `szEP2SrcIPUDP'
Testlib.c:82: warning: unused variable `szEP2DstIPUDP'
Testlib.c:86: warning: unused variable `VopEvents'
Testlib.c:89: warning: unused variable `chan'
Testlib.c:90: warning: unused variable `ch'
Testlib.c:91: warning: unused variable `fCard1'
Testlib.c:94: warning: unused variable `nNumIPChan'
Testlib.c:95: warning: unused variable `IPStr'
Testlib.c:97: warning: unused variable `wAttrib'
Testlib.c:97: warning: unused variable `wPayloadType'
Testlib.c:97: warning: unused variable `wVIFSize'
gcc -Wall -g -c parse_args.c -o parse_args.o
parse_args.c: In function `parse_args':
parse_args.c:70: warning: implicit declaration of function `atoi'
gcc -g -o Testlib Testlib.o parse_args.o -L. -lVoIP
[root@localhost testlib]#

Figure 6-8 *Rebuilding Testlib Series Programs*

6.2.1 Rebuilding the SigC5561 Terminal Emulation Program

Following are instructions for rebuilding the SigC5561 terminal emulation program (see also section 4.2 above, PCI Terminal Program).

1) Log in as root
2) Change working directory to be:

```
/home/Signalogic/SigC5561/Test_Programs/terminal
```

Build the terminal emulation program by executing the following commands:

```
make clean
make Terminal
```

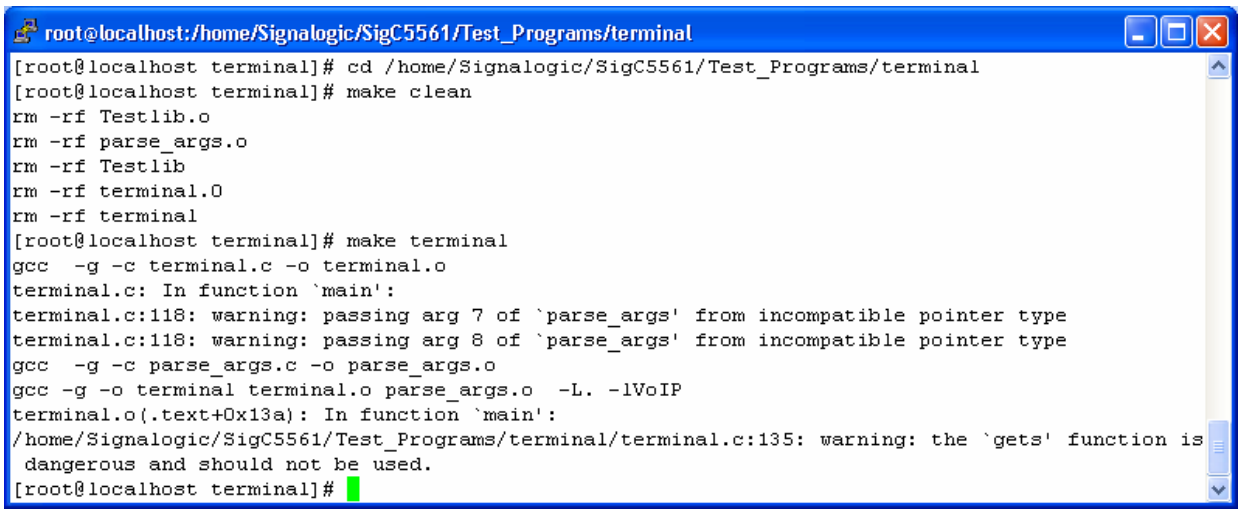
root@localhost:/home/Signalogic/SigC5561/Test_Programs/terminal
[root@localhost terminal]# cd /home/Signalogic/SigC5561/Test_Programs/terminal
[root@localhost terminal]# make clean
rm -rf Testlib.o
rm -rf parse_args.o
rm -rf Testlib
rm -rf terminal.O
rm -rf terminal
[root@localhost terminal]# make terminal
gcc -g -c terminal.c -o terminal.o
terminal.c: In function `main':
terminal.c:118: warning: passing arg 7 of `parse_args' from incompatible pointer type
terminal.c:118: warning: passing arg 8 of `parse_args' from incompatible pointer type
gcc -g -c parse_args.c -o parse_args.o
gcc -g -o terminal terminal.o parse_args.o -L. -lVoIP
terminal.o(.text+0x13a): In function `main':
/home/Signalogic/SigC5561/Test_Programs/terminal/terminal.c:135: warning: the `gets' function is
dangerous and should not be used.
[root@localhost terminal]#

Figure 6-9 *Rebuilding the SigC5561 Terminal Emulation Program*

6.3 TDM-to-IP Testing

Below is an overview diagram outlining data and signaling flow for the TDM-to-IP test software covered in this section.

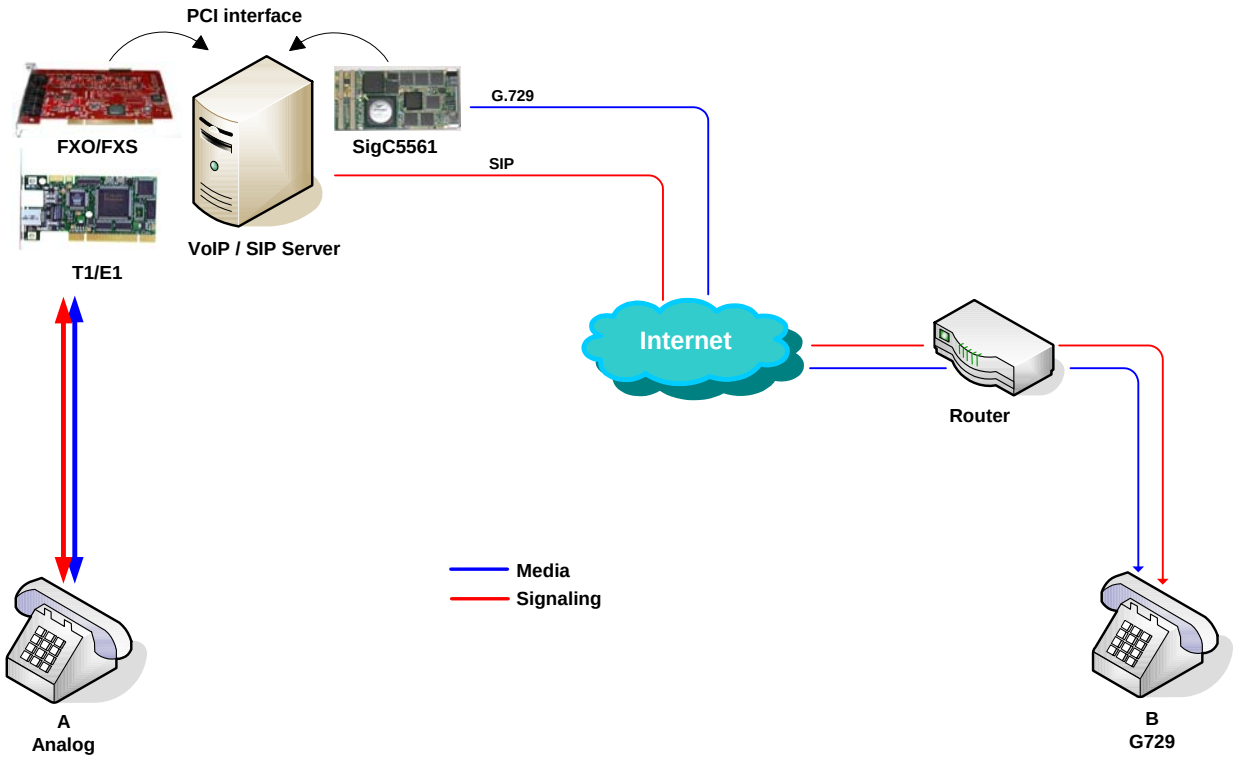

Figure 6-10 *SigC5561 PCI Card TDM-to-IP Test Setup*

The above diagram is general in that it shows both a multi-port FXO/FXS card and a multi-port T1/E1 card to handle analog telephony data, commonly referred to as TDM data when moved across the PCI bus. However, in real situations typically both would not be present in the system (although certain situations may warrant that). Another possible telephony hardware option would be a card that supports E&M signaling.

Figure 6-11 below contains a more detailed diagram (including routing information) showing the SigC5561 card in a LAN situation, behind a router.

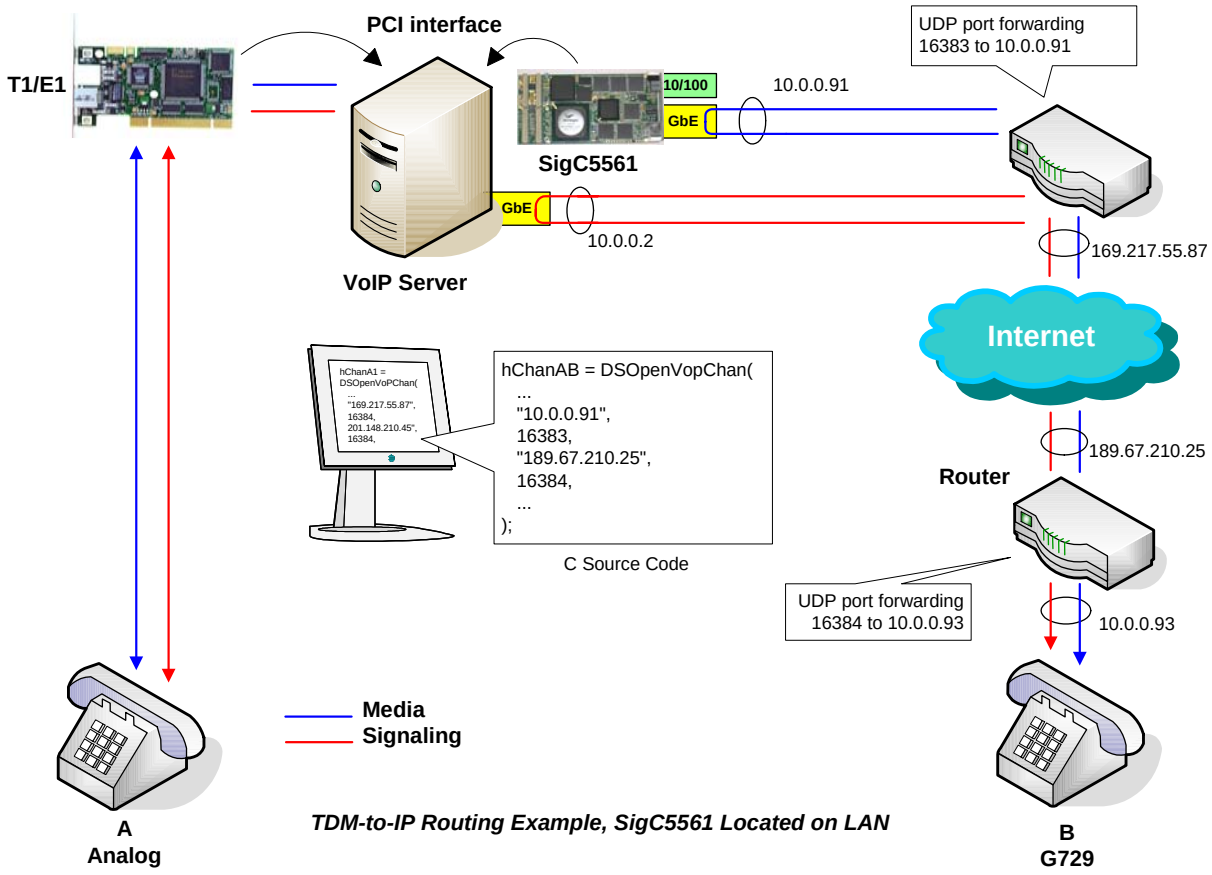

Figure 6-11 *SigC5561 PCI Card LAN Routing, TDM-to-IP Operation*

In Figure 6-11 above, note the UDP port forwarding required at the VoIP Server and equivalent use of local source IP addresses in the following C source code excerpt:

```
hChanAB = DSOpenVopChan(
                        ...
                        "10.0.0.91",
                        16383,
                        "187.67.210.25",
                        16384,
                        ...
                       );
```

Figure 6-12 below contains a more detailed diagram showing the SigC5561 card in a WAN situation, positioned directly on the public network.

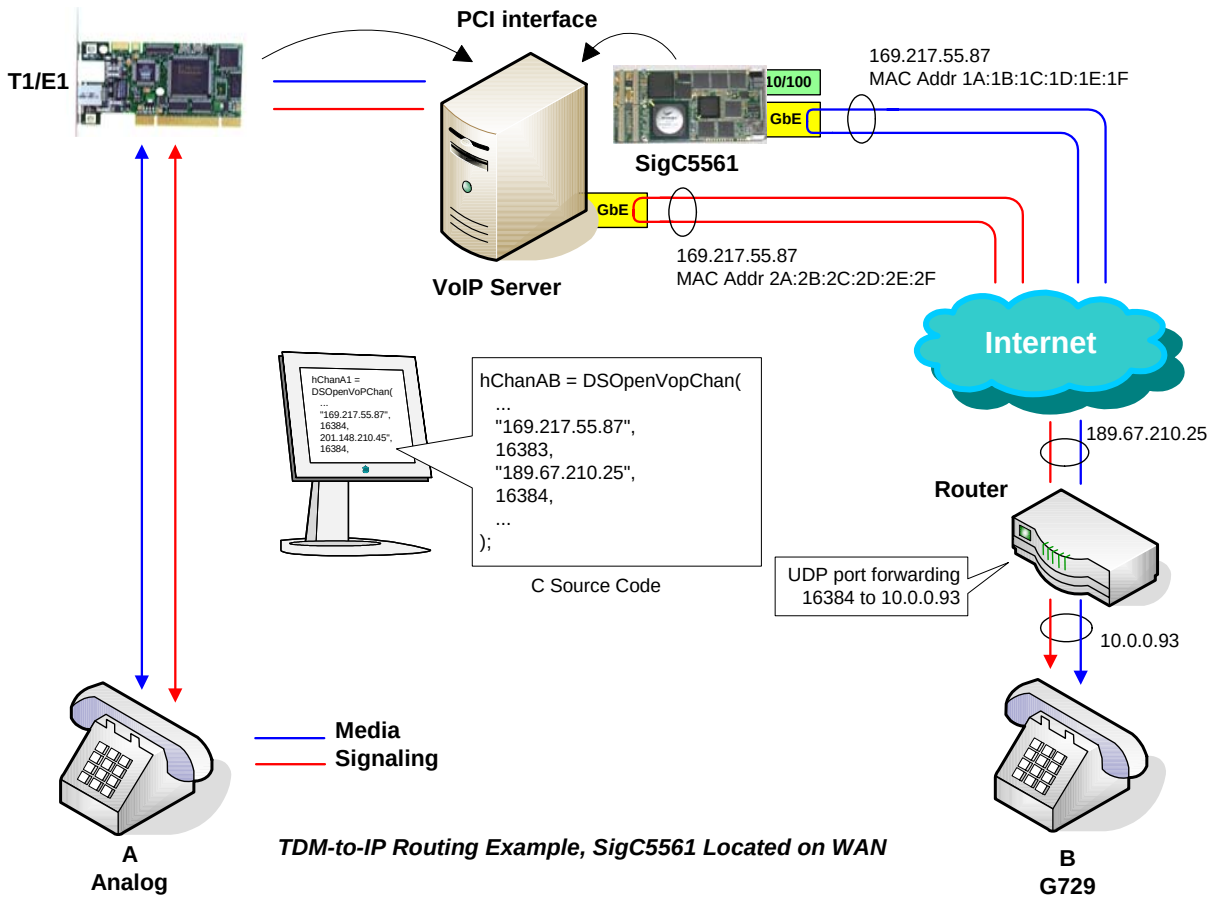

Figure 6-12 *SigC5561 PCI Card WAN Routing, TDM-to-IP Configuration*

In Figure 6-12 above, note the UDP port forwarding required at the destination end-points and use of public IP addresses in the following C source code excerpt:

```
hChanAB = DSOpenVopChan(
                          ...
                          "169.217.55.87",
                          16383,
                          "189.67.210.25",
                          16384,
                          ...
                         );
```

To run the TDM-to-IP test program, type the following command-line:

```
./TestlibTDM2IP -ips10.0.0.92 -ipd10.0.0.93 -n10 dnld -v
```

This will open 10 channels of TDM-to-G729 VoIP between the specified source and destination IP addresses; i.e. endpoints A and B in the diagram in Figure 6-11 and Figure 6-12 above. You should see a display similar to image shown in Figure 6-13 below:

 Copyright © Signalogic 2006-2007

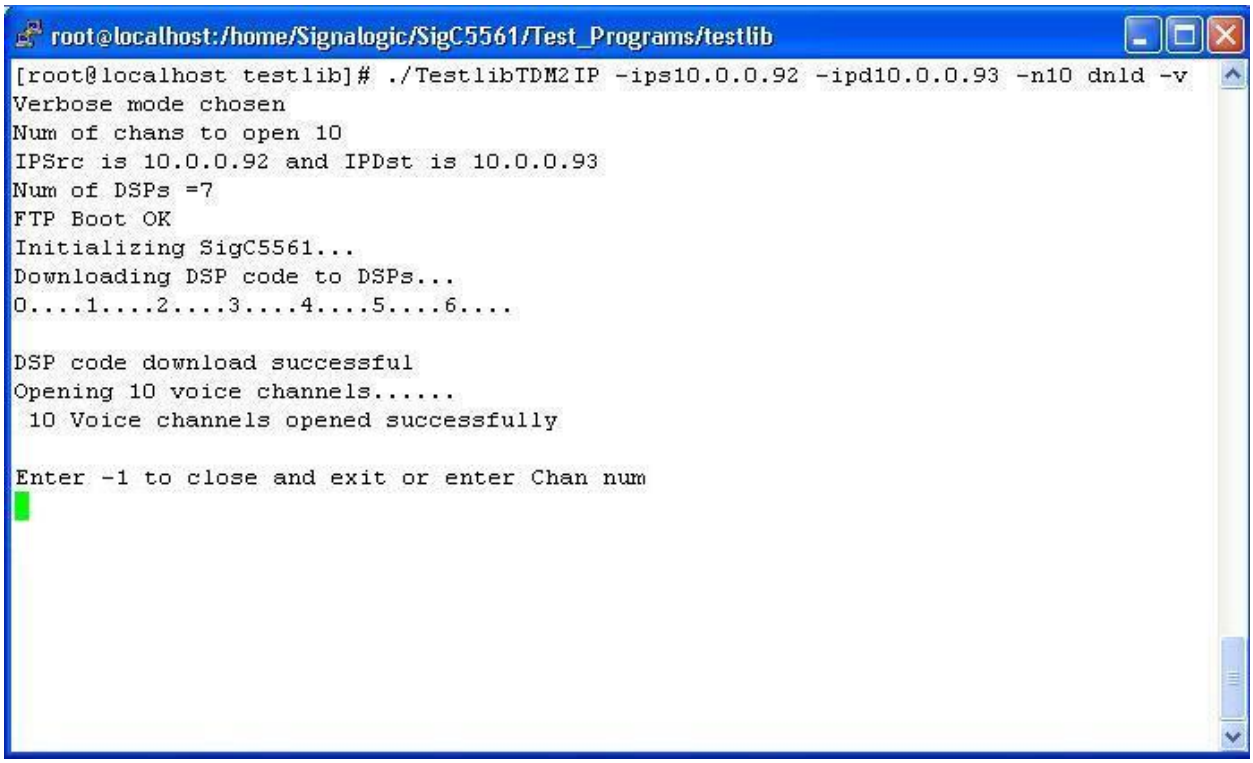
root@localhost:/home/Signalogic/SigC5561/Test_Programs/testlib
[root@localhost testlib]# ./TestlibTDM2IP -ips10.0.0.92 -ipd10.0.0.93 -n10 dnld -v
Verbose mode chosen
Num of chans to open 10
IPSrc is 10.0.0.92 and IPDst is 10.0.0.93
Num of DSPs =7
FTP Boot OK
Initializing SigC5561...
Downloading DSP code to DSPs...
0....1....2....3....4....5....6....
DSP code download successful
Opening 10 voice channels......
10 Voice channels opened successfully
Enter -1 to close and exit or enter Chan num

Figure 6-13 *TestlibTDM2IP Initial Display*

Note the above example shows 7 DSPs being downloaded, allowing up to 672 TDM-to-G.729 voice channels. A 2-DSP card would be limited to 192 channels.

6.4 IP-to-IP Testing

Below is a diagram outlining basic data and signaling flow for the IP-to-IP test software covered in this section.

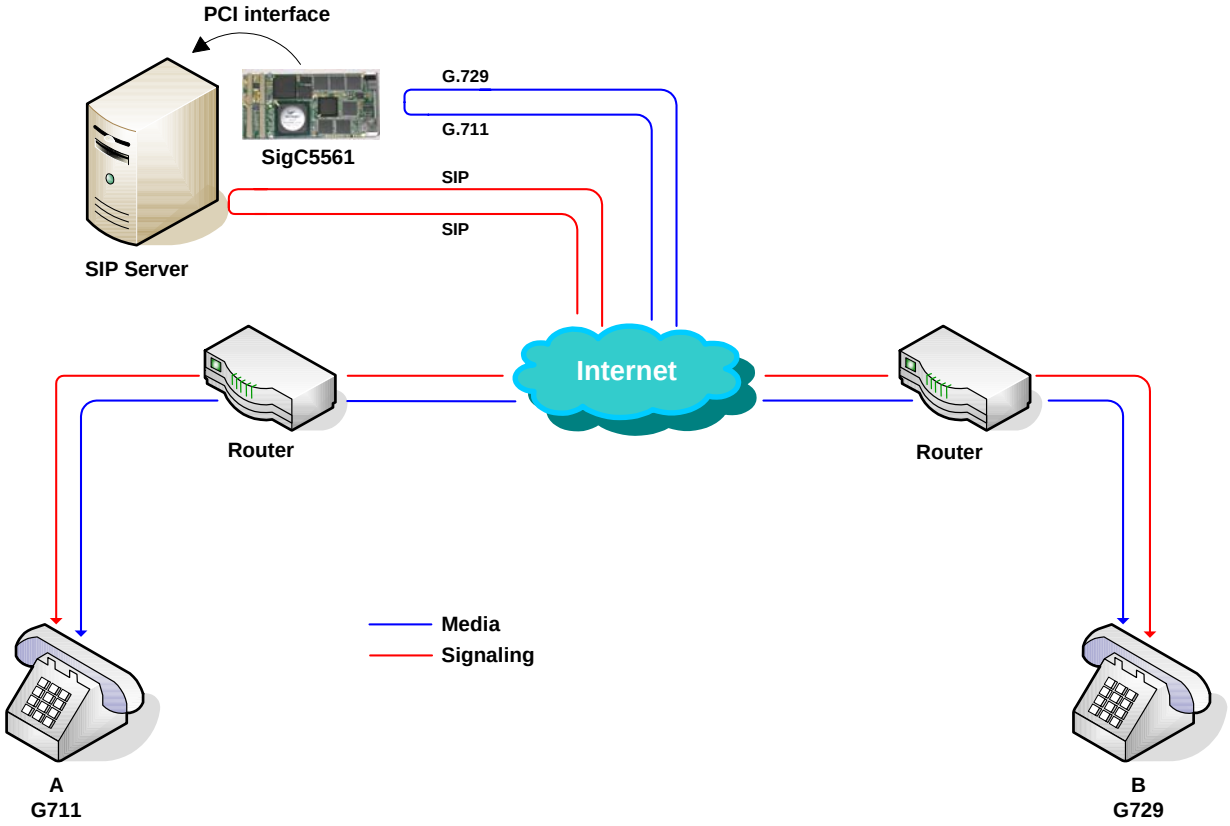

Figure 6-14 *SigC5561 PCI Card IP-to-IP Test Setup*

Figure 6-15 below contains a more detailed diagram (including routing information) showing the SigC5561 card in a LAN situation, behind a router.

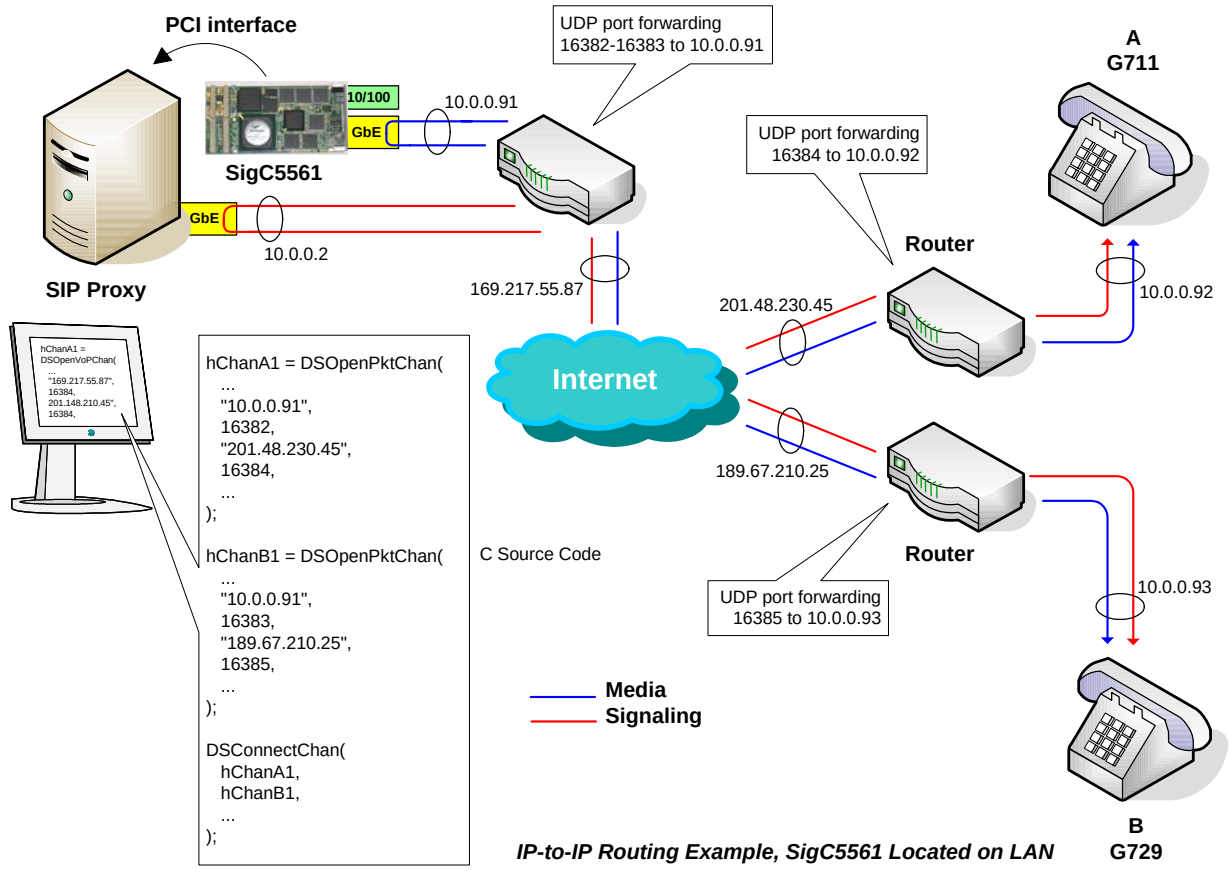

Figure 6-15 *SigC5561 PCI Card LAN Routing, IP-to-IP Configuration*

In Figure 6-15 above, note the UDP port forwarding required at the SIP Proxy and equivalent use of local source IP addresses in the following C source code excerpt:

```
hChanA1 = DSOpenPktChan(
                        ...
                        "10.0.0.91",
                        16382,
                        "201.48.230.45",
                        16384,
                        ...
                       );

hChanB1 = DSOpenPktChan(
                        ...
                        "10.0.0.91",
                        16383,
                        "189.67.210.25",
```

```
                        16385,
                        ...
                       );

DSConnectChan(
                hChanA1,
                hChanB1,
                ...
            );
```

Figure 6-16 below contains a more detailed diagram showing the SigC5561 card in a WAN situation, positioned directly on the public network.

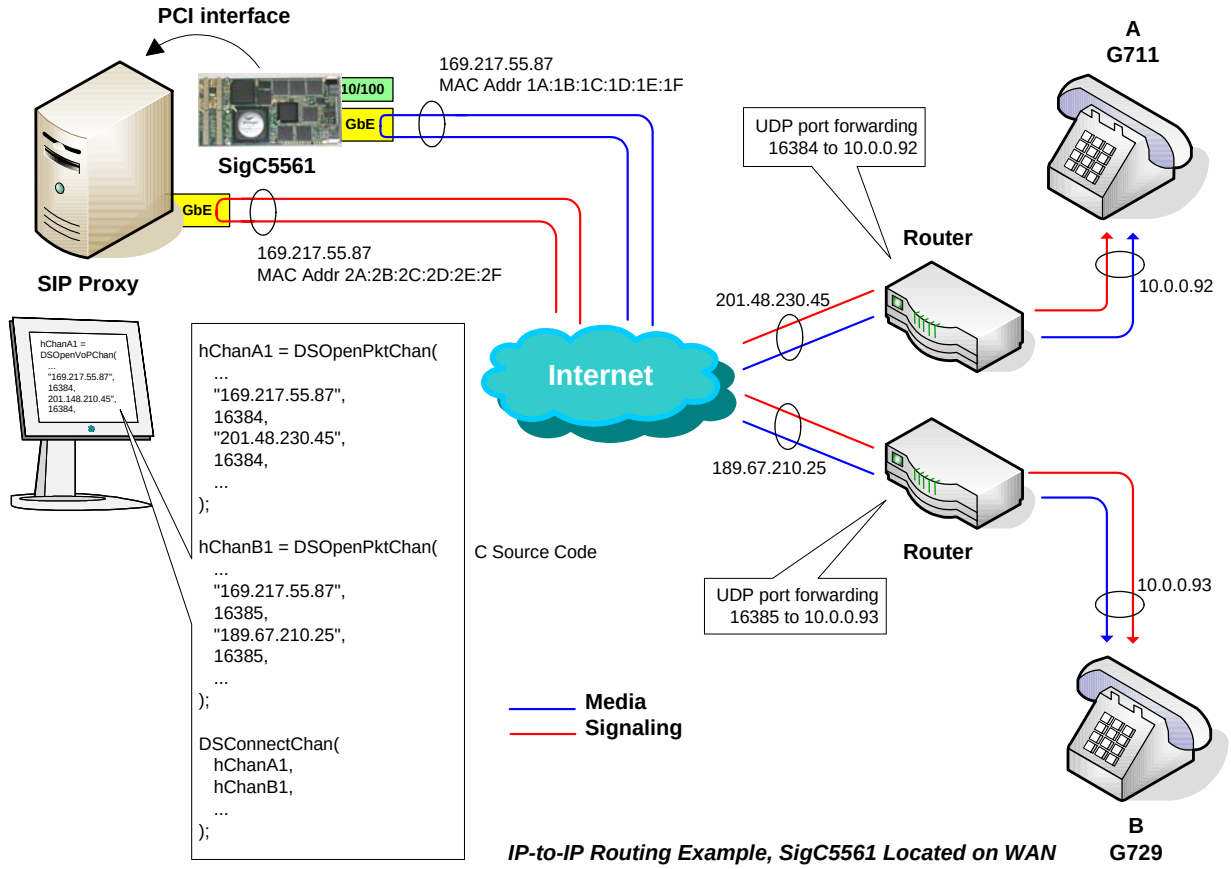

Figure 6-16 *SigC5561 PCI Card WAN Routing, IP-to-IP Configuration*

In Figure 6-16 above, note the UDP port forwarding required at the destination end-points and use of public IP addresses in the following C source code excerpt:

```
hChanA1 = DSOpenPktChan(
                        ...
                       "169.217.55.87",
```

```
                          16384,
                          "201.48.230.45",
                          16384,
                          ...
                         );

  hChanB1 = DSOpenPktChan(
                          ...
                          "169.217.55.87",
                          16385,
                          "189.67.210.25",
                          16385,
                          ...
                         );

  DSConnectChan(
                 hChanA1,
                 hChanB1,
                 ...
               );
```

To run the IP-to-IP test program, type the following command-line:

```
./TestlibIP2IP -ips10.0.0.92 -ipd10.0.0.93 -n10 dnld -v
```

This will open 10 channels of G729-to-G711 transcoding between the specified source and destination IP addresses; i.e. endpoints A and B in the diagram in Figure 6-15 and Figure 6-16 above. You should see a display similar to image shown in Figure 6-17 below:

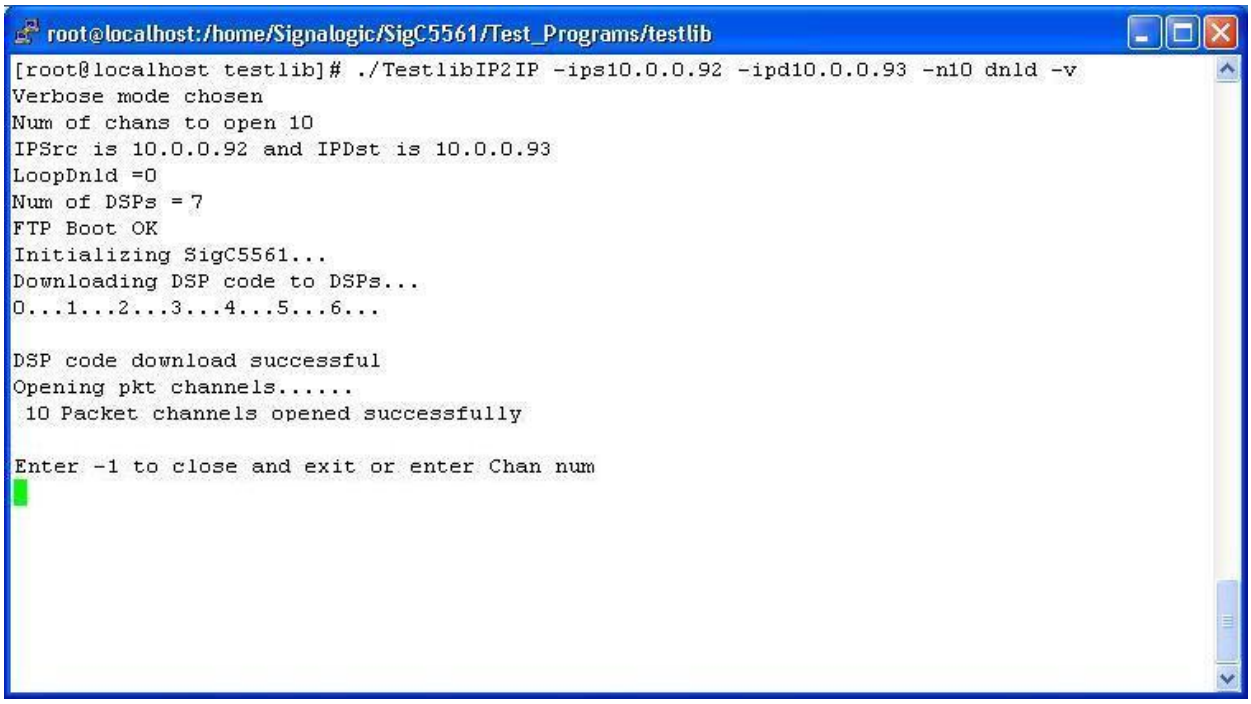
root@localhost:/home/Signalogic/SigC5561/Test_Programs/testlib
[root@localhost testlib]# ./TestlibIP2IP -ips10.0.0.92 -ipd10.0.0.93 -n10 dnld -v
Verbose mode chosen
Num of chans to open 10
IPSrc is 10.0.0.92 and IPDst is 10.0.0.93
LoopDnld =0
Num of DSPs = 7
FTP Boot OK
Initializing SigC5561...
Downloading DSP code to DSPs...
0...1...2...3...4...5...6...
DSP code download successful
Opening pkt channels......
10 Packet channels opened successfully
Enter -1 to close and exit or enter Chan num

Figure 6-17 *TestlibIP2IP Initial Display*

Note the above example shows 7 DSPs being downloaded, allowing up to 672 G.711-to-G.729 channel transcoding. A 2-DSP card would be limited to 192 channels.

 Copyright © Signalogic 2006-2007

6.5 Conferencing Example

*** Under Construction ***.

6.6 Asterisk SIP Proxy Example

A configuration showing Asterisk running in combination with a SIP proxy on a VoIP server to form a softswitch is shown in Figure 6-18 below. In this scenario "softswitch" includes both a media gateway (endpoints A, B, and C) and IP-to-IP transcoding services (endpoints B and C).

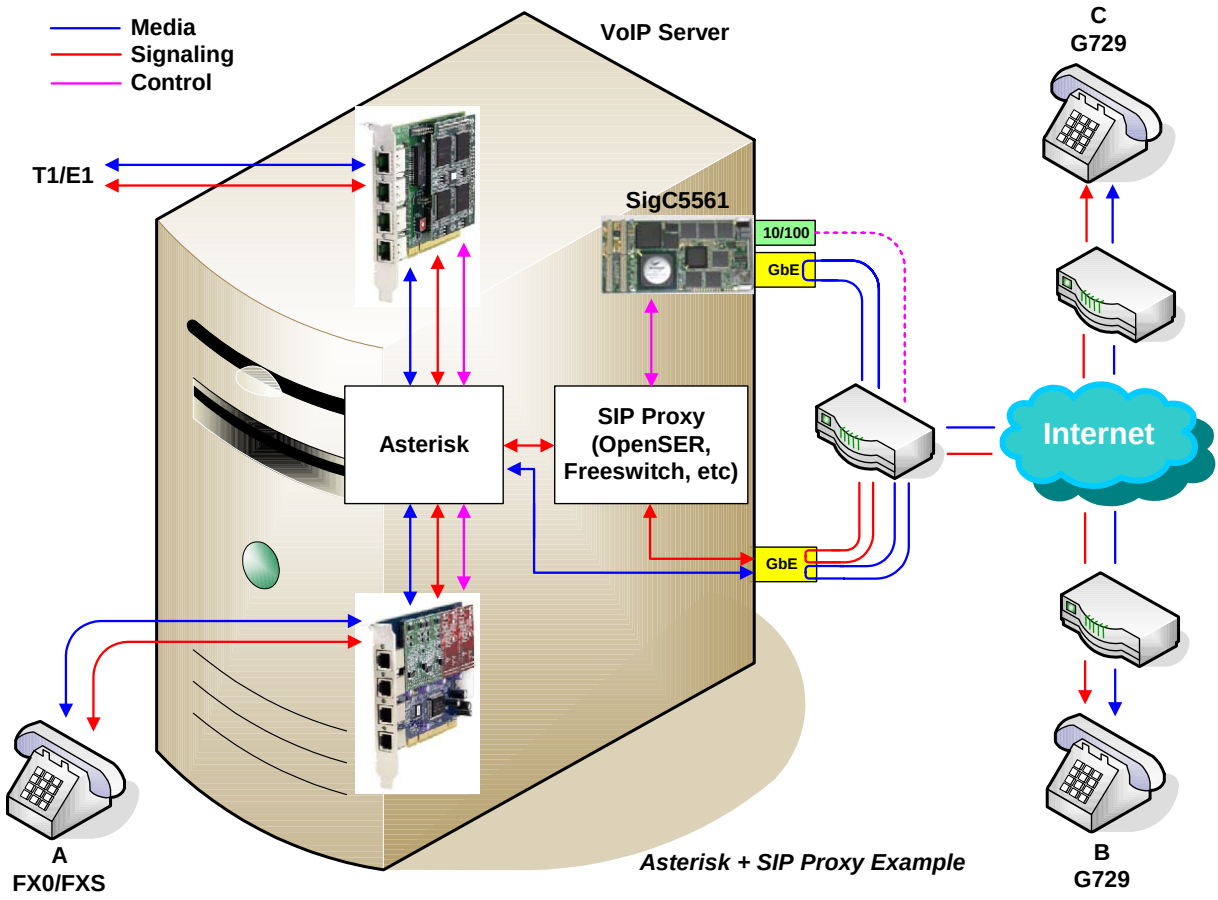

Figure 6-18 *Asterisk TDM Termination Combined with SIP Proxy*

SigC5561 documentation includes the above configuration instead of running Asterisk alone, in back-to-back user agent (B2BUA) configuration, for two (2) reasons:

1. Software packages other than Asterisk are more efficient and more stable for high channel capacity SIP processing
2. Asterisk open source has not been modified to allow B2BUA operation with a high capacity hardware resource such as the SigC5561, which includes onboard IP/UDP/RTP packetization and GbE termination

Regarding item 2 above, Asterisk architecture does not envision a situation where TDM or intermediate transcode data is sent to a hardware resource and "never seen again". Instead

Asterisk always expects voice data "to return" in the form of RTP packets across the PCI bus for further handling. In addition, B2BUA operation in Asterisk is complex and involves several software modules -- maintaining the required patches as Asterisk itself changes over time is likely to be problematic.

Figure 6-19 below shows TDM-to-IP voice data flow in a combined Asterisk and SIP proxy system. A commonly used example would be G.729 encoding on the IP side.

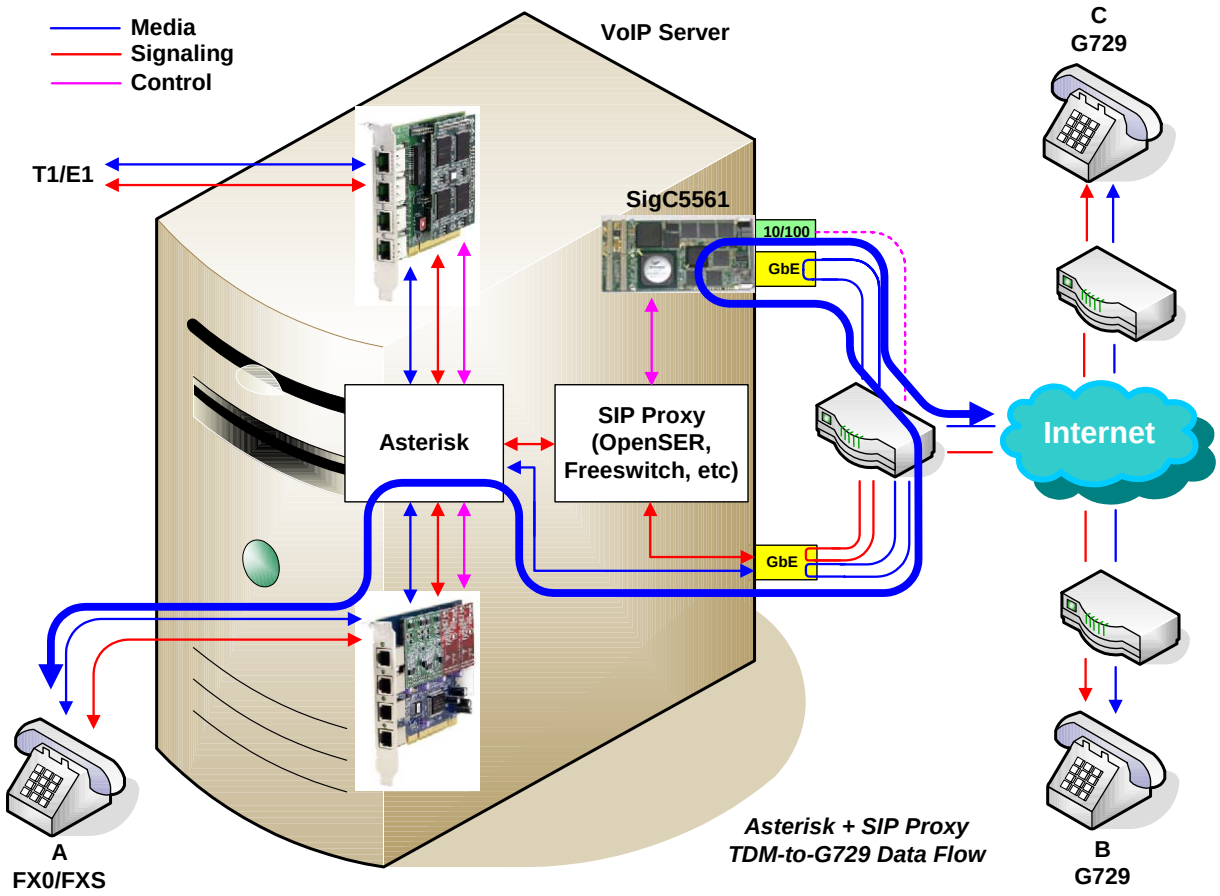

Figure 6-19 *TDM-to-G729 Data Flow in Asterisk + SIP Proxy System*

Note that in the above example, Asterisk operates unaware of G.729 (or other codec) encoding requirements at the IP-side endpoint, simply moving G.711 data over IP – it's most efficient, stable mode. Voice data is then G.729 encoded by the SigC5561 card, without impact on host HMP resources, and routed by the SigC5561 card to its final destination.

Note also that echo cancellation may be performed either by the FXS/FXO or T1/E1 cards (if they offer that feature), Asterisk software, or the SigC5561 card. The SigC5561 card offers the highest channel capacity and highest quality echo cancellation choice, consuming no host HMP resources and utilizing carrier grade Telogy algorithms. The Telogy echo canceller has been developed over many years .by Texas Instruments, with literally millions of dollars invested in

constant improvement in order to satisfy clients operating in a wide range of environments, including Cisco, Nortel, and Lucent-Alcatel.

Figure 6-20 below shows IP-to-IP voice data flow in a combined Asterisk + SIP proxy system.

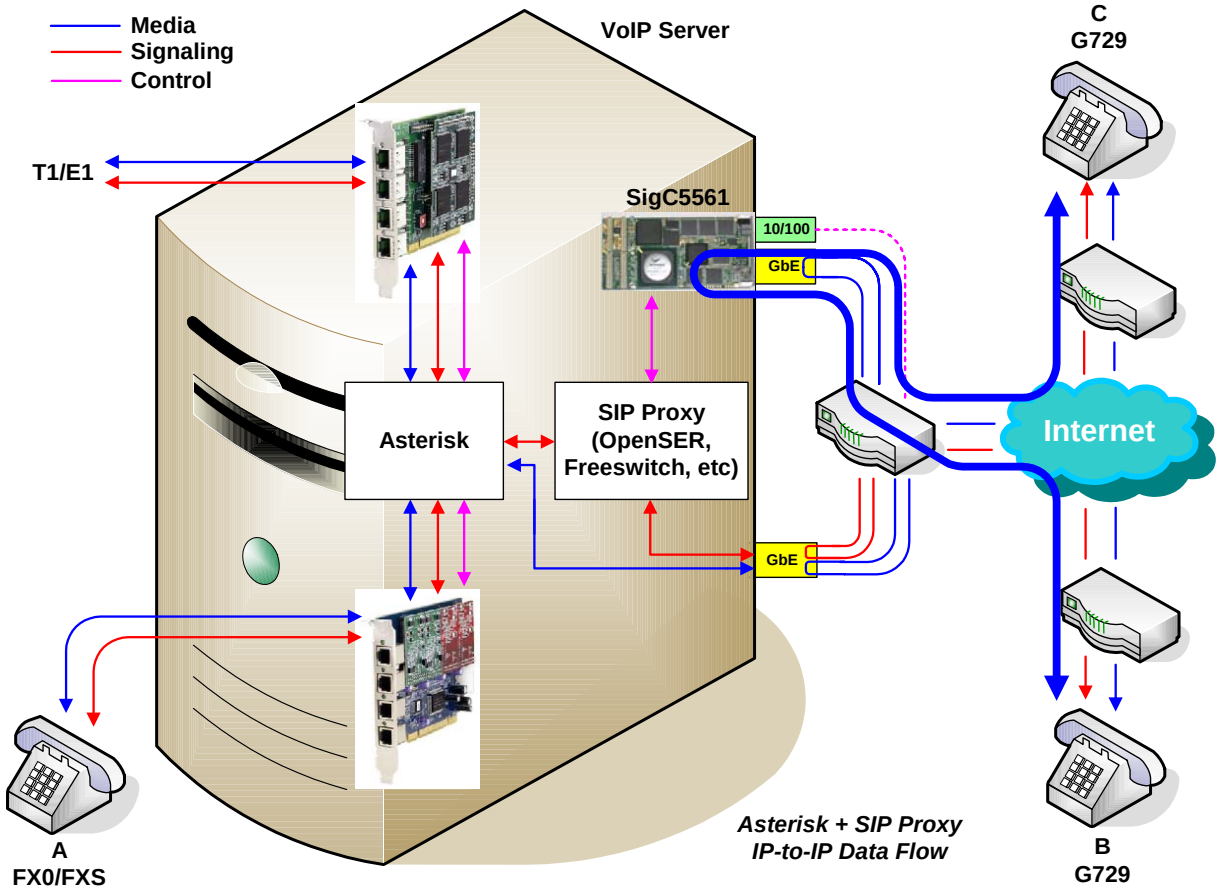

Figure 6-20 *IP-to-IP Data Flow in Asterisk + SIP Proxy System*

Note that in the above example, Asterisk is not involved unless PBX functions such as IVR, announcement, caller ID, etc. are required.

 Copyright © Signalogic 2006-2007

7 Card Resources and Peripherals

7.1 Host Interface

The host interface allows an external or host processor to “see” and communicate with onboard DSPs, logic, and peripherals.

The SigC5561 PCI card and SigC5561-PTMC module both implement a standard PCI v2.2 compliant interface, explained in more detail in the following sections.

 Copyright © Signalogic 2006-2007

7.1.1 PCI Host Interface

Host software “sees” the PCI host interface as three (3) separate memory-mapped areas:

- “high performance” BAR that allows high data rate burst mode and bus master transfers to one or more DSPs
- “buffer” BAR area used for burst mode and bus master transfers
- “compatibility mode” BAR (Base Address Register) that includes all register sets available in the PC104 Host Interface (see section 10.1.1 above)

An overall view of DSP processor cores, logic registers, and peripheral registers accessible via the PCI host interface is shown below in Figure 7-1.

 Copyright © Signalogic 2006-2007

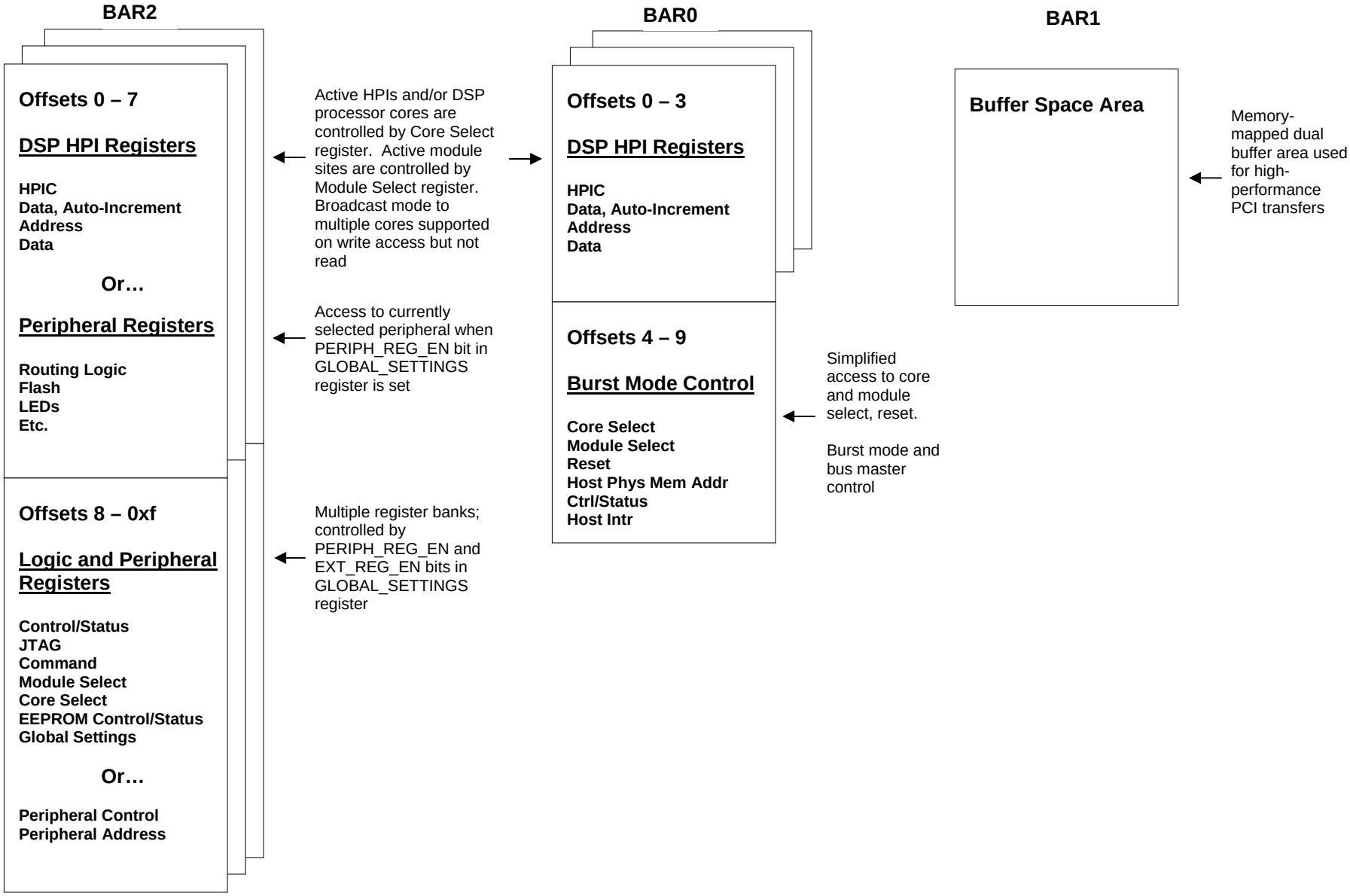

Figure 7-1 *PCI Host Interface Overview*

Host PCI Interface Notes

1. All register offsets use 32-bit addressing.
2. BAR2 registers are typically 8 and 16 bits wide; BAR0 and BAR 1 registers are 32-bits wide.

7.1.1.1 PCI Configuration Space Values

Below are current PCI configuration space values for SigC5xxx and SigC6xxx PCI boards. Note that this information is subject to change.

Field	Value	Comments
Vendor ID [1]	0x10EE	Xilinx PCI core logic value
Device ID [1]	0x5190	Xilinx PCI core logic value
Class Code	0x0B40	Generic Co-Processor
Rev ID	0x0090	Logic revision v0.90
Subsystem Vendor ID	0x10EE	Vendor ID repeated
Subsystem Device ID	0x5190	Subsystem Device ID, repeated

Notes

[1] Temporary value, subject to change.

7.1.1.2 BAR0 Description

Following are PCI register definitions for BAR0:

Register Name	32-Bit Offset	DSP Accessible	Comments
HPIC	0	Y	HPI Control/Status register
HPIA	1	Y	HPI Address
HPIDI	2	N	HPI Data, with auto-increment
HPID	3	N	HPI Data, no auto-increment
CTRL_STATUS	4	N/A	Control/Status register
MODSEL	5	N/A	Module Select register
CSEL	6	N/A	Core/Chip Select register
CSEL_EX1	7	N/A	Core/Chip Select register extension 1
CSEL_EX2	8	N/A	Core/Chip Select register extension 2

RESET	9	N/A	Reset register
DMA_HOST_ADDR	10	N/A	Host memory physical address register
DMA_HOST_CTRL	11	N/A	DMA control register, host side
DMA_DSP_CTRL	12	N/A	DMA control register, DSP side
DMA_FLAGS	13	N/A	DMA flags and status register
INTR_STATUS	14	N/A	Interrupt register

HPIC, HPIA, HPIDI, and HPID Registers

The HPIC, HPIA, HPIDI, and HPID registers map to HPI registers inside each DSP. When one or more CSEL bits for DSPs are active, the host driver has direct access to HPI of the active DSPs.

Only single-PCI transfer access is permitted to HPI registers. Access to these registers stalls the PCI bus while the EDMA of the selected DSP services the transaction. If more than one DSP is active, then the PCI bus must wait for the slowest EDMA.

For more information on HPI registers, see section 2.3, TMS320C64x HPI16 or HPI32, in the Texas Instruments document TMS320C6000 DSP Host Port Interface (HPI) Reference Guide (TI document number SPRU578B).

CTRL_STATUS Register

RESERVED (31 -- 8)	X	HHWIL	CORE_RST	PROC_RST	INT3	INT1	INT0	X

The Ctrl/Status register contains different DSP control and status bits. Only single-PCI transfer access is permitted to this register. Ctrl/Status register bits are defined as follows:

INT0

DSP INT0 enable. Currently not implemented.

INT1

DSP INT1 enable. Currently not implemented.

INT3

DSP INT3 enable. Currently not implemented.

PROC_RST

Processor Reset bit. When the PROC_RST bit is set (or cleared), the edge change causes ones (or zeros) to latch into the RESET register for DSPs that have their corresponding bits set in the CSEL register. If the DSP's CSEL bit is zero, there would be no change in its Reset register bit.

See below for more information on the RESET and CSEL registers.

CORE_RST

Core Reset bit. Currently not implemented.

HHWIL. Currently not implemented.

MODSEL Register

RESERVED (31 -- 8)	M7 – M0

The Module Select register allows groups of DSPs to be enabled/disabled. Some boards support actual physical "module sockets", other boards support banks of DSPs. When an Mn bit is set, all operations on the DSPs in that module is permitted, including core select, memory data transfer, reset, etc. When an Mn bit is clear, the DSPs in the module are held in Reset and no DSP operations are permitted.

Only single-PCI transfer access is permitted to this register.

Notes

1. For SigC641x PCI cards or PTMC modules, only M0 is active. M1-M7 are reserved and should not be accessed, but if written then written as zero.

2. For SigC5561 PCI cards or PTMC modules, only M0 and M1 are active (each bank contains up to 6 DSPs). M2-M7 are reserved and should not be accessed, but if written then written as zero.

CSEL Register

C31	C30 -- C3	C2	C1	C0

CSEL registers use bit fields to select which DSP HPI ports are currently active. Memory access and reset register operations use the current CSEL setting to determine on which cores on which devices to operate.

A '1' bit indicates a core is active during memory access, reset, and other processor operations, and a '0' indicates a core is not active.

Only single-PCI transfer access is permitted to this register.

Notes

1. Each Cn bit corresponds to a chip-select signal. For SigC641x PCI cards or PTMC modules, only C0-C7 are active. C8-C95 are reserved and should not be accessed, but if written then written as zero.

2. For SigC5561 PCI cards or PTMC modules, only C0-C11 are active. C12-C95 are reserved and should not be accessed, but if written then written as zero.

3. Multi-core DSP operation is not implemented in logic versions 0.9x or earlier.

4. Internally, the logic does not allow multiple chip-select signals to be active during HPI reads. If software driver calls are made such that more than one chip-select signal is active at the time of HPI read, the logic activates only the chip-select corresponding to the first set lsb in the CSEL registers. For example, a CSEL register setting of 0x000A would reduce to 0x0002.

RESET Register

RST31	RST30 -- RST3	RST2	RST1	RST0

The Reset Register allows direct control of Reset lines for individual DSPs. Clearing a bit will place the corresponding DSP into Reset, and setting a bit will release the DSP from Reset.

Only single-PCI transfer access is permitted to this register.

Notes

3. For SigC641x PCI cards or PTMC modules, only RST0-RST7 are active. RST8-RST31 are reserved and should not be accessed, but if written then written as zero.

4. For SigC5561 PCI cards or PTMC modules, only RST0-RST11 are active. RST12-RST31 are reserved and should not be accessed, but if written then written as zero.

 Copyright © Signalogic 2006-2007

DMA_HOST_ADDR Register

HA31 – HA0

After host application layer software allocates a memory buffer to receive bus-mastering transfer data from the SigC5/6xxx board, it must pass the memory buffer "logical address" to the host driver. The driver must then determine the actual host system memory "physical address" and make it known to SigC5/6xxx board PCI logic, by setting the DMA_HOST_ADDR. This must be done prior subsequent bus mastering transfer, initiated by writing to the DMA_CTRL register.

Only single-PCI transfer access is permitted to this register.

DMA_HOST_CTRL Register

DIR	BUFSEL	RESERVED (29 – 10)	LEN (9 – 0)

The DMA_HOST_CTRL register allows the host driver to control and initiate bus-mastering transfers. Writes to the register initiate a bus-mastering transfer with the control values written.

Only single-PCI transfer access is permitted to this register.

Following are descriptions of individual DMA_HOST_CTRL register bits.

DIR

DIR is the direction indicator. If DIR is cleared, the SigC5/6xxx board will transfer data from the host system memory to its local transfer buffer. If DIR is set, the SigC5/6xxx board will transfer data from its local buffer into host system memory.

BUFSEL

BUFSEL is the buffer selection indicator. If BUFSEL is cleared, the transfer will be to/from Buffer0 inside the SigC5/6xxx board's local transfer buffer memory area. If BUFSEL is set, the transfer will use Buffer1.

LEN

LEN specifies the bus master transfer length as the number of 32-bit words. The current maximum transfer length is 1024.

 Copyright © Signalogic 2006-2007

DMA_DSP_CTRL Register

DIR	BUFSEL	RESERVED (29 – 10)	LEN (9 – 0)

The DMA_DSP_CTRL register allows DSP code to control and initiate bus-mastering transfers. Writes to the register by a DSP initiatee a bus-mastering transfer with the control values written.

Following are descriptions of individual DMA_DSP_CTRL register bits.

DMA_FLAGS Register

PE	SSE	RMA	RTA	STA	DST	DPE	FBBC	0	66MHZ	CL	IS	RSRVD (18 – 8)	MA	TA	DWD	DWOD	TT	NT	TE	DT

SigC5/6xxx board PCI interface logic can detect differing methods of transaction termination including no-error. The DMA_FLAGS register provides access to those detection codes, serving as driver integrity and diagnostic tool.

Only single-PCI transfer access is permitted to this register.

Following are DMA_FLAGS register bit definitions; all values are active high (1 == active) and given from the perspective of SigC5/6xxx board PCI logic.

PE

Parity Error Detected

SSE

Signaled System Error

RMA

Received Master Abort

RTA

Received Target Abort

STA

Signaled Target Abort

DST

 Copyright © Signalogic 2006-2007

Device Select Timing

[10] –Slow
[01] – Medium

DPED

Data Parity Error Detected

FBBC

Fast Back-to-Back Capable, Always 0

Bit22

Always 0

66MHZ

66 MHz Capable, Always 0

CL

Capabilities List

IS

Interrupt Status

MA

Master Abort

TA

Target Abort

DWD

Disconnect With Data

DWOD

Disconnect Without Data

TT

Target Termination

NT

Normal Termination

TE

Transaction End

DT

Data Transfer

INTR_FLAGS Register

RESERVED (31 – 1)	BMTC

The Interrupt Flags register allows SigC5/6xxx logic to signal interrupts to the host driver by setting bits on a per-interrupt basis. The host driver may clear an interrupt by writing a '1' to the corresponding bit.

After reacting to a SigC5/6xxx board interrupt, the host driver must clear the corresponding interrupt bit to prevent further, indefinite interrupts from occurring. The host driver may read the INTR_FLAGS register value and write it back to clear all devices asserting interrupts.

Only single-PCI transfer access is permitted to this register.

Following are bits defined in the INTR_FLAGS register:

BMTC

Interrupt indicating that bus mastering transfer has completed.

7.1.1.3 BAR1 Description

BAR1 maps buffer space used for high-performance mode PCI transfers to/from one or more DSP HPI(s).

Byte offset 0x0000 – 0x0FFF is defined as Buffer0 and 0x1000 – 0x1FFF is as Buffer1.

 Copyright © Signalogic 2006-2007

7.1.1.4 BAR2 Description

BAR2 memory map follow the register offset and functional definition given in sections 10.2 through 10.4 below. BAR2 provides a “compatibility mode” interface to allow existing software utilities, diagnostic and instrumentation and measurement software to function with only a driver change. Examples include register-level board test suite and Hypersignal-Macro software.

7.2 DSP HPI Register Access

DSP HPI access from host and from DSP software is enabled via logic interface on the hardware. From the base I/O address (ISA bus) or BAR0 (PCI bus), the first 8 locations correspond to HPI registers in the C5xxx or C6xxx device. Note that registers are byte-addressable to support C5xxx device HPI8 access, but also may be accessed as words on word boundary to support HPI16 device access.

For more information on PCI bus access, see section 10.1.2 above, PCI Interface.

Note that for DSP devices which are target-only (e.g. HPI port only, with no bus master capability), a DMA controller implemented in logic is used to allow DSP devices to access board peripherals, external host bus, and other DSP devices via their HPI ports.

Example host and DSP C code identifiers for DSP HPI accesses are given below.

C Code Identifier	**Offset**	**Comments**
DSP_CTRL_REG_HI	0	DSP HPIC register high byte. See TI documentation for detailed information. Access as byte for C5xxx devices in HPI8 mode, access as word for C5xxx and C6xxx devices in HPI16 mode.
DSP_CTRL_REG_LO	1	DSP HPIC register low byte. See TI documentation for detailed information. Access as byte for C5xxx devices in HPI8 mode, ***do not access*** for C5xxx and C6xxx devices in HPI16 mode.
DSP_DATA_REG_HI_AINC	2	DSP data auto-increment register high byte. See TI documentation for detailed information. Access as byte for C5xxx devices in HPI8 mode, access as word for C5xxx and C6xxx devices in HPI16 mode.
DSP_DATA_REG_LO_AINC	3	DSP data auto-increment register low byte. See TI documentation for detailed information. Access as byte for C5xxx devices in HPI8 mode, ***do not access*** for C5xxx and C6xxx devices in HPI16 mode.
DSP_ADDR_REG_HI	4	DSP address register high byte. See TI documentation for detailed information. Access as byte for C5xxx devices in HPI8 mode, access as word for C5xxx and C6xxx devices in HPI16 mode.

 Copyright © Signalogic 2006-2007

DSP_ADDR_REG_LO	5	DSP address register low byte. See TI documentation for detailed information. Access as byte for C5xxx devices in HPI8 mode, ***do not access*** for C5xxx and C6xxx devices in HPI16 mode.
DSP_DATA_REG_HI	6	DSP data register high byte. See TI documentation for detailed information. Access as byte for C5xxx devices in HPI8 mode, access as word for C5xxx and C6xxx devices in HPI16 mode.
DSP_DATA_REG_LO	7	DSP data register low byte. See TI documentation for detailed information. Access as byte for C5xxx devices in HPI8 mode, ***do not access*** for C5xxx and C6xxx devices in HPI16 mode.

Detailed information about these registers can be found in Texas Instruments C54xx, C55xx, and C6xxx series documentation, under sections labeled HPI and Enhanced HPI. Below is a list of Texas Instruments books in which to look:

C54x CPU and Peripherals
C54x Enhanced Peripherals
C6000 Peripherals

 Copyright © Signalogic 2006-2007

7.3 Logic Registers

Logic register access from host and from DSP software is enabled via logic interface on the hardware. From the base I/O address (ISA bus) or BAR0 (PCI bus), the second 8 locations (8 through 15) are mapped to logic registers when the PERIPH_REG_EN bit in the GLOBAL_SETTINGS register is set to 0.

Note that registers are byte-addressable, and in some cases accessible as words on word boundary, in order to support both byte-addressable and word-addressable peripherals.

Note that for DSP devices which are target-only (e.g. HPI port only), a DMA controller is used to allow DSP access to board peripherals, external host bus, and other DSP device HPI ports.

Below is additional information about logic registers.

Logic Registers

Host and Module Interface

Offset †	Main Registers	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8	CTRL_STATUS [1]	DMA_ACTIVE	SYS_RESET_N	CORE_RESET_N ‡	PROC_RESET_N	INT0	INT1_BIO	INT3	JTAG_SEL
9	JTAG_CTRL_STATUS	LPBK_EN	TMS	TCK	TDI	TDO	RSRVD	RSRVD	OUTPUT_EN ‡
0xa	COMMAND [2]	M_INT_REG_EN	HI_BYTE_EN	RESERVED					
0xb	MODULE_SEL [3]	MOD8_EN	MOD7_EN	MOD6_EN	MOD5_EN	MOD4_EN	MOD3_EN	MOD2_EN	MOD1_EN
0xc	CORE_SEL_LO [2, 4]	CORE7_EN	CORE6_EN	CORE5_EN	CORE4_EN	CORE3_EN	CORE2_N	CORE1_EN	CORE0_EN
0xd	CORE_SEL_HI [2, 4]								
0xe	EE_I2C_DEVICE [6]	EE_BOOT_STAT	RESERVED			SER_DATA_OUT	SER_DATA_IN	DEVICE_CS	SER_CLK
0xf	GLOBAL_SETTINGS [5]	PERIPH_REG_EN	DSP_8/16 [7]	HOST_8/16 [7]	DMA_EN	RESERVED	INT1_BIO_EN [7]	INT3_EN [7]	EXT_REG_EN [8]

General Notes

† Offset is relative to base I/O address (ISA bus) or BAR (PCI bus). All offsets assume 16 available locations.

‡ "_N" is used to denote an active low signal. "EN" is used as an abbreviation for "Enable".

All bits are read/write unless otherwise denoted by (RO) or (WO) designators. Registers with "STATUS" suffix contain one or more hardware inputs that may be read and monitored.

All register bits not defined are reserved and should be written as a zero. Undefined bits may not read as zero.

Register Specific Notes

[1] For modules with on-module logic, CTRL_STATUS and CORE_SEL registers are reflected in on-module logic. In this case, writes cause both baseboard and module logic registers to be updated, reads access only module logic registers.

[2] COMMAND register bits 5-0 are reserved and should be left unused if legacy hardware must be supported.

[3] MODULE_SEL register enables one or more module sites or DSP banks, depending on the board type.

[4] CORE_SEL register bits select processors for boards/modules with single-core devices. CORE_SEL register is 16-bits deep when HOST_8/16 is zero. The upper 8 bits may be accessed by odd byte reads when HOST_8/16 is 1. CORE_SEL register is 32-bit deep when EXT_REG_EN bit is active; used for example on SigC544x boards.

[5] GLOBAL_SETTINGS register visible when bit 7 (PERIPH_REG_EN) is zero; Peripheral Register bank is visible when bit 7 is one.

[6] EE_I2C_DEVICE allows host and DSP interface to I2C bus on board, or other serial bus devices, such as serial EEPROM. In some cases other EEPROM devices use this register; EE_BOOT_STAT provides indication of valid Flash EEPROM boot.

[7] See Reset Conditions notes.

[8] EXT_REG_EN on some boards enables additional registers.

Reset Conditions

DSP_8/16 and HOST_8/16 are set hy module type auto-detection logic at power-up, as follows:

Module Type	DSP_8/16	HOST_8/16	INT3_EN	INT1_BIO_EN
No Module	1	1	1	0
C549 (legacy)	1	1	0	0
C54xx	1	1	0	0
C55xx	0	1	0	0
C6xxx	0	1	0	0

SigC54/55/6xx Logic Register Definitions

Rev H2, May 2004
Copyright (C) Signalogic, 2002-2004

Revision History

Created: Apr 2002
Revised: Sep 2002, added peripheral control
Revised: Nov 2002, T1/E1 and DUART additions
Revised: Jan 2003, C6xxx additions
Revised: Jun/Jul 2003, Flash and I/O enables, misc updates
Revised: Aug 2003, additional notes and corrections
Revised: Nov 2003, digital I/O and Flash EEPROM corrections
Revised: Dec/Jan 2004, External Bus Master Support
Revised: Mar 2004, TDM Routing Logic Improvements
Revised: Apr 2004, Added Power Control register, made LED register banks a peripheral
Revised: May 2004, added EE_BOOT_STAT, INT1_BIO_EN, SYS_RESET

 Copyright © Signalogic 2003-2004

Logic Register Notes

1. For modules with on-module logic, CTRL_STATUS and CORE_SEL registers are reflected in on-module logic. In this case, writes cause both baseboard and module logic registers to be updated, reads access only module logic registers. Examples of this case include the SigC67xx and SigC5561 module series. An example of a module design with no on-module logic is the SigC54xx module series.

2. COMMAND register bits 5-0 are reserved and should be left unused if legacy hardware must be supported.

3. MODULE_SEL register enables one or more module sites or DSP banks, depending on the board type.

4. CORE_SEL register bits select processors for boards/modules with single-core devices. CORE_SEL register is 16-bits deep when HOST_8/16 is zero. The upper 8 bits may be accessed by odd byte reads when HOST_8/16 is 1. CORE_SEL register is 32-bit deep when EXT_REG_EN bit is active; used for example on SigC544x boards.

5. GLOBAL_SETTINGS register visible when bit 7 (PERIPH_REG_EN) is zero; Peripheral Register bank is visible when bit 7 is one.

6. The EE_I2C_DEVICE register allows host and DSP interface to a PC104 or PTMC board I2C bus, or other serial bus devices, such as serial EEPROM. In some cases other EEPROM devices use this register; as an example, the EE_BOOT_STAT bit provides indication of valid Flash EEPROM boot.

7. See Reset Conditions notes in Logic Registers graphic above.

8. EXT_REG_EN on some boards enables additional registers. See board-specific documentation.

Specific Register Notes

Below are additional notes about specific registers.

Ctrl/Status Register

1) Default value of the Control/Status register at power-up or system reset is dependent on module type auto-detection logic that executes at logic boot-time, and also on whether the stand-alone mode jumper (JP14 on PC/104 boards) is installed. Depending on the module type installed, possible power-on / system reset values of the Control/Status register are shown below:

 Copyright © Signalogic 2006-2007

Module Type	**Value if Stand-Alone Mode Jumper Installed**	**Value if Stand-Alone Mode Jumper Not Installed**
No Module	0	0
C549 (legacy)	0x1e	0
C5409A	0x1e	0
C5416	0x1e	0
C5421	0x3e	0
C55xx	0x3e	0
C6xxx	0x3e	0

2) DSP processor Reset signals are maintained by internal registers that are not directly accessible to host or DSP software, but are indirectly accessible. Internal reset registers are latched ***only on transitions*** of the Reset bit in the Control/Status register, and depending on current value of the Core Select register. For example, to reset one or more processors, the following sequence should be followed:

- set Core Select Register (see below) to zero; no processors are selected
- write 1 to PROC_RESET_N bit in Control/Status Register (and/or CORE_SELECT_N bit if DSP is multi-core device)
- select desired processors in Core Select Register
- write 0 to Reset bit in Control/Status Register

After the above sequence, the selected processors will be held in Reset.

3) Processors may be released from the reset state as needed, either all at once or in any combination, as needed. The same procedure as above should be followed, except sequence of writes to the PROC_RESET_N bit should be 0-1.

4) It is recommended to hold unused processors in reset, to both minimize power consumption on the module and to avoid possible situations where a processor runs undefined code and adversely affects other processors.

5) For multicore devices such as C5421 and C5441 the PROC_RESET_N bit applies to HPI Reset signals and the CORE_RESET_N bit is used for individual cores. For single-core C5xxx devices, only the PROC_RESET_N bit is active.

Module Select Register

1) Default value of the Module Select Register at power-up is zero. Module type auto-detection logic that runs at logic boot-time may set this register to 1.

2) Each bit in the Module Select Register enables a module site on the board, as shown above. More than one module may be selected at one time.

3) For DSP Farm type boards without module sites, each bank of 8 DSPs is considered a "module". There are a maximum of 32 cores in a module or bank.

Core Select Register

1) Default value of the Core Select Register at power-up is zero. Module type auto-detection logic that runs at logic boot-time may set this register to 1.

2) For modules with multicore processors each bit represents a core. For example, the SigC5421 module contains eight (8) C5421 processors with two (2) cores each; in this case, the Core Select Register is 16 bits wide and each bit represents a core.

3) For modules with on-module CPLD logic, this register is "reflected" inside the CPLD logic and actually exists there, and not in the PC104 or PTMC board logic.

4) For modules with less than N processors or cores, bits 31 .. N-1 in the Core Select Register are not used.

5) Each bit in the Core Select Register enables a processor in currently selected module site(s). Multiple processors may be selected at once; in this case the standard Signalogic host software drivers can perform "broadcast" writes. ***Read operations should only be attempted with one processor selected at a time to avoid contention between DSP HPI ports.***

Global Settings Register

1) Default value of the Global Settings register at power-up or system reset is dependent on module type auto-detection logic that executes at logic boot-time. Depending on the module type installed, possible power-on / system reset values of the Global Settings register are shown below:

Module Type	Value
No Module	0x62
C549 (legacy)	0x62
C54xx	0x60
C55xx	0x20
C6xxx	0x20

2) The PERIPH_REG_EN bit enables onboard peripherals to be enabled at addresses base + 0 to base + 7, instead of DSP modules. See section 10.4 below, Peripheral Registers.

 Copyright © Signalogic 2006-2007

3) DSP_8/16 determines whether DSP module accesses are 16-bit ore 8-bit. Default value is set according to module type auto-detection logic; see table above.

4) Host_8/16 determines whether host bus accesses are handled as 8-bit or 16-bit. Default value is set according to module type auto-detection logic; see table above. This bit must be set prior to making 16-bit register accesses from host software drivers.

5) DMA_EN enables the DMA controller logic in test/debug mode. This bit should not be set by customers under any circumstances.

6) CLK_DIV enables division by 2 of the master logic clock for internal usage. This bit is used for test and special modes by Signalogic and should not be set by customers under any circumstances.

Example host and DSP C code identifiers for logic register accesses are given below. When developing DSP code applications, see the file rtc_periph.h for more information.

C Code Identifier	**Offset**	**Comments**
CTRL_STATUS_REG	8	Control/Status Register
JTAG_REG	9	JTAG Register
CMD_REG	0xa	Command Register
MOD_SEL_REG	0xb	Module Select Register
CORE_SEL_REG	0xc	Core Select Register
GLOBAL_SETTINGS_REG	0xf	Global Settings Register

 Copyright © Signalogic 2006-2007

7.4 TDM Registers

This section not yet documented.

Below is additional information about peripheral registers.

 Copyright © Signalogic 2003-2004

This section not yet documented.

 Copyright © Signalogic 2003-2004

TDM Register Notes

This section not yet documented.

7.5 Flash

SigC5561 products contain two types of Flash:

1. NOR (boot) Flash
2. NAND (media) Flash

The boot Flash is used to store boot code for the WinPath network processor, including low-level chip configuration and VxWorks BSP (board support package) configuration. The boot Flash may optionally also hold the main WinPath code image (the default is FTP download at boot-time).

7.5.1 Boot Flash Configuration

Prior to shipment, the SigC5561 PCI card or PTMC module is flashed with VxWorks configuration. This low-level configuration code and data acts essentially as a BIOS for the card. If a system (with Linux Operating System) is provided for software installation at Signalogic, there should already be a directory called "/home/Signalogic/SigC5561/Utilities/flash". Otherwise, please check with Signalogic if your Linux system kernel version is supported. However, following are steps to flash the SigC5561 card if there should be updates available for the low-level boot configuration code and data.

1. Determine the SDRAM (U14 and U15) memory size on the SigC5561 card before flashing the VxWorks configuration. Small memory size of 256 KB has part marking of "**D9CBP**". Big memory size of 512 KB has part marking of "**D9DDQ**".
2. Make sure the SigC5561 card is installed into a 3.3V PCI slot on the mother board. Reference the **Section 2.2** above for installing the card into the system.
3. Power on the system.
4. After Linux has loaded successfully, open up a command terminal and go to /home/Signalogic/SigC5561/Drivers/. Load the driver if not loaded earlier using the 'make load' command. Please refer to the driver sub section below on how to build and load the driver.
5. Go to "/home/Signalogic/SigC5561/Utilities/flash".
6. Execute the flash programming utility as shown below. Please use the command line './flash_vxworks –m16' instead of '-m8' if the marking on the U14, U15 devices is "D9DDQ".

 Copyright © Signalogic 2006-2007

Figure 7-2 *'flash' Command for Smaller Size 256 KB SDRAM Memory*

7. A successful flashing of the SigC5561 card is shown in Figure 7-3 below.

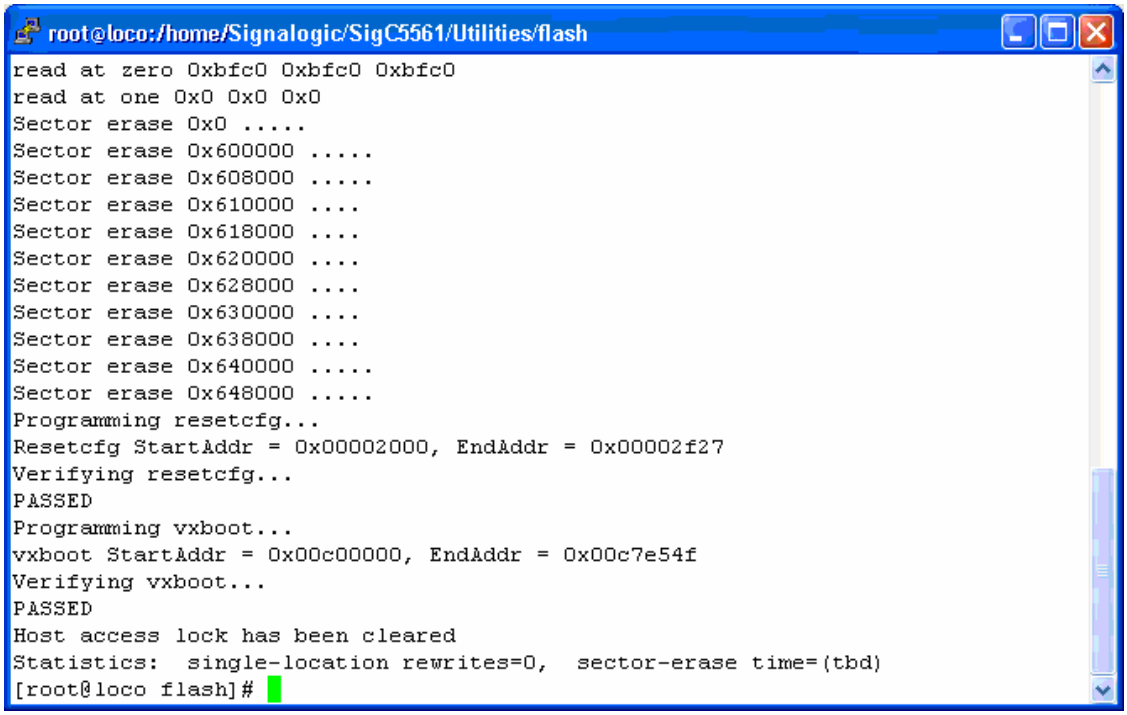

Figure 7-3 *'flash' Command Successful Result*

 Copyright © Signalogic 2006-2007

7.5.2 Network Processor Code Flash Configuration

*** This section under construction ***

 Copyright © Signalogic 2006-2007

7.6 LEDs

The SigC5561 PCI card and SigC5561-PTMC module contain the following LEDs:

1. front-panel (bezel) LEDs for status of DSP, PCI, IP
2. front-panel (bezel) LEDs for status of
3. row of 16 small “status and error code” LEDs on top PCB surface, near the front-panel

7.6.1 Panel LED Indicators

7.6.2 Status / Error Code LED Indicators

When LED_CTRL bits are zero LEDs will reflect system values, as shown below:

LED	**Description of System Value**	**Comments**
LED0 (CR1 on board silkscreen)	Indicates logic boot status	1 for boot Ok, 0 or flashing if error
LED1 (CR2 on board silkscreen)	Indicates stand-alone mode initialization status	1 for stand-alone initialization Ok, 0 or flashing if not

When LED_CTRL bits are 1, LEDs will reflect user values. In this case, LED_DATAn values determine LED state, under user control. A 1 value for bit n will illuminate the LEDn. See note 1 above.

7.7 DMA Controller

The DMA controller provides a method of moving data between DSPs and peripherals without host processor intervention. For PTMC modules, the DMA controller is also actively involved in burst mode and bus master transfers.

The DMA controller can be accessed both by host software and by DSP software. For DSP software, it provides the main method of allowing one or more DSPs to transfer data to/from other DSPs and to access onboard peripherals.

For host software, the DMA controller is less often used; for PTMC modules DMA controller functions are utilized transparently, depending on the type of API-level call made by host application software, and what driver-level functions that are invoked as a result. From both host and DSP perspectives, control register bits are accessible in the CTRL_STATUS and GLOBAL_SETTINGS registers.

7.7.1 DMA Controller Testing

7.7.2 Initial DMA Configuration

In DSP code, an initial DMA controller configuration statement is required at the start of void main(), as shown below:

```
SetDevice(RTC_DEV_ALL, RTC_DEV_ENABLE_RELEASE | DEV_DISABLE);
```

The above line is a good idea for any situation where JTAG debug is being used. It disables all I/O peripherals and releases any "enable hold" pending internally to the peripheral I/O library. The intent is to force the peripheral library -- and to some extent DMA controller logic -- to a known state regardless of what point JTAG emulation was stopped during the previous debug session.

7.7.3 DMA Controller Performance

The logic currently handles burst transfers between DSP and DUART, routing logic, and Flash -- burst mode, or transferring multiple values per transaction, is by far the most efficient.

In the example test code, the ReadDigitalIOData() and WriteDigitalIOData() function calls mentioned in section 3 above are currently invoked in "synchronous mode", where they wait for the DMA transaction to complete. All six (6) DMA transactions (digial I/O accesses) inside the real-time frame loop in melp_rt example C code require about 360 usec total in synchronous mode. This time is spent by the peripheral library waiting to return the caller (blocking), which may eventually cause an issue in a real-time system that is tight on MIPS resources. To make transactions faster, asynchronous mode can be used, which causes the peripheral library to not wait for DMA transactions to complete. When asynchronous mode is active, *only one* call to

 Copyright © Signalogic 2006-2007

Read or Write should be made during each frame loop, so there should be a counter that increments from 0 to 5 and starts over. For example:

```
/* assumes count, bank2_io, bank3_io, and bank4_io are set to zero */

   if (count == 0)
      bank2_io = ReadDigitalIOData(RTC_IO_BANK2) & 0x18;
   else if (count == 1)
      bank3_io = ReadDigitalIOData(RTC_IO_BANK3) & 0x0f;
   else if (count == 2)
      bank4_io = ReadDigitalIOData(RTC_IO_BANK4) & 0xe0;
   else if (count == 3)
      WriteDigitalIOData(RTC_IO_BANK2, bank2_io);
   else if (count == 4)
      WriteDigitalIOData(RTC_IO_BANK3, bank3_io);
   else if (count == 5)
      WriteDigitalIOData(RTC_IO_BANK4, bank4_io);

   count++;
   if (count == 5) count = 0;
```

The advantage in this situation is that each ReadDigitalIOData() or WriteDigitalIOData() function call only requires less than 500 nsec. However, panel I/O latency becomes dependent on the real-time frame loop period and the number of digital I/O accesses. In the above structure, there would be an approximate 135 msec delay to update all panel I/O when running at 2400 bps and an approximate 400 msec delay when running at 1200 bps. A latency less than 1/2 second should be acceptable for a user-interface; if not, the above code can be made "smarter" to adjust the count variable such that the I/O bank needing update (output), as a result of user input, is accessed during the next frame loop. It should be possible to always maintain a latency under 70 msec using that approach.

Also, for future system consideration, a version of DMA controller logic is under development that will allow a single call to ReadDigitalIOData() and WriteDigitalIOData() to update from 1 to 4 digital I/O banks at one time. This will improve the performance of synchronous mode operation, as well as simplifying application level code to some extent.